FUNCTION POINT METRIC CALCULATION AND SENSITIVITY ANALYSIS

By

TONGCHIT TANTIKUL

Bachelor of Science

in Statistics

Chulalongkorn University

Bangkok, Thailand

1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May 1998

FUNCTION POINT METRIC CALCULATION
AND SENSITIVITY ANALYSIS

Thesis Approved

Mansur Samadzadeh

Thesis Advisor

H. Lu

Wayne B Powell

Dean of the Graduate College

PREFACE

Managing software during development and maintenance is one of the most troubling aspects of modern information technology. Documented historical records indicate that poorly managed projects can run into cost overruns and schedule slippages and, in some cases, may be abandoned. Careful attention to quality control by using trained specialists, modern software estimation tools, and effective planning tools can immunize software development companies against some of the common sources of software disaster. The software development process can thus be managed more thoroughly or even controlled.

The size of a software document at various stages of development can be measured using a number of alternative methods. Lines of code is probably the size measure most widely known, but it has no universally accepted definition. Function points is one of the better-known and widely-used metrics for measuring software size.

The objective of this thesis was to study and analyze the sensitivity of the function point metric. The work done consists of collecting the size in terms of the number of Lines of Code (LOC) of various programs, converting the lines of code to function points, calculating function points from the program documentation independently of the previous calculation, comparing the estimated function points and the actual one, and investigating the sensitivity of function point parameters by plotting a graph. The Line of

Code was measured by the CSIZE software tool, and the function points was calculated by the COSMOS software tool. It was found that among all the function point parameters (i.e., external inputs, external outputs, internal logical files, external interface files, and external inquiries), the internal logical files is more sensitive than the rest.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to and thank my thesis advisor Dr. Mansur H. Samadzadeh, who motivated me by his valuable instruction and example throughout my thesis research work. Dr. Samadzadeh gave generously of his valuable time and expertise for correcting my work and rendering helpful suggestions.

I also extend my sincere appreciation to Drs. Huizhu Lu and John Hatcliff for their advice and willingness to serve on my graduate committee. Their suggestions and support were very helpful throughout the study.

Also, I would like to express my gratitude to my parents, sister, and all my friends who were mentally with me all the time.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

Software is inescapable in the modern world. It has become the key element in all kinds of systems such as transportation, telecommunication, medicine, entertainment, education, military, and engineering. The rise in demand for software has become a critical problem. Most of the time, the problem occurs because of a lack of understanding of the field of software engineering [Jones 96b].

The field of software engineering has existed for the past four decades, and has been defined by IEEE [Pressman 97] as follows.

> Software engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

The cost of software has continued to increase, while the cost of hardware has continued to decrease dramatically [Sommerville 96] [Boehm 81]. The growing cost of software has focused the attention of the computing community on the measurement and estimation aspects of the software development process.

One of the goals of software estimation is to provide software managers and users

with reasonably accurate answers to the following questions about a software project [Putnam 80].

Can we do it?
How long will it take?
How much will it cost?
What is the risk?
What is the trade-off?

There are a number of ways to find answers to the above questions. For example, the COCOMO methodology and the associated software tool have been used to estimate the software development effort (person-month), and the function point method has been used to estimate the number of Lines of Code (LOC).

Function point metric is one of the fastest growing software measurement techniques in the software industry [Bennatan 95]. Function point measures the size of software based on the functionality that the users request and receive. Since this metric is based on a number of empirical parameters, experience is an important factor in the successful calculation and application of function point estimation [Low and Jeffery 90].

The main objective of this thesis was to study the function points metric and analyze its sensitivity with respect to the various factors, stages, and parameters involved in its calculation. It was found that among all the function point parameters (i.e., external inputs, external outputs, logical internal files, external interface files, and external inquiries), logical internal files is the most sensitive parameter.

The organization of this thesis is as follows. Chapter II of this thesis report provides a literature review and the overview of various software complexity metrics. Chapter III describes function point calculation in detail. Chapter IV explains the

experimental methodology and sensitivity analysis of function points. Chapter V contains the summary of the thesis and some suggestions for future work.

CHAPTER II

SOFTWARE COMPLEXITY

Researchers give different definitions for software complexity. Fenton defines complexity as a term to capture all internal attributes of software [Fenton 91]. Software complexity metrics can be used as indicators of various aspects of software products that provide the information necessary to control projects. For instance, they can be used to measure the size, modularity, or control-flow of a program.

An important aim of software development is to reduce overall software costs. Software project managers must be able to make estimates of how much a particular software development, or part of that development, is going to cost.

Accurate cost estimation is important to software managers for several reasons. For example, one can consider the making of a decision as to whether or not to proceed with a software development, in bidding on a contract or quoting a price to a customer, and the need to avoid substantial cost overruns or underruns. Overestimating can result in failure to win a contract, while underestimating can result in a loss to the new development effort or a very unhappy customer [McDermid 94].

Early software cost estimations were based principally on expert judgment or, where it was available, historical information for software development projects similar to the development project being estimated.

In the 1970's, a number of algorithmic software cost estimation models began to emerge. The most popular use of software complexity to estimate software cost are COCOMO (Constructive Cost Model), SLIM (Software Life Cycle Model), and the Function Points Model.

2.1 SLIM

The Software Life Cycle Model (SLIM) estimation method was developed in 1978 by Larry Putnam [Putnam 80]. SLIM uses the Rayleigh curve model to describe the effort estimates. The Rayleigh curve is a well-known exponentially declining curve. In SLIM the software equation for estimates the size of the software measured by LOC is of the form [Kemerer 87]:

$$size\ (LOC) = CK^{\frac{1}{3}} t_d^{\frac{4}{3}}$$

where

size: number of delivered source instructions (LOC),

K: life cycle effort in person years,

C: a technology constant that depends on use of programming techniques, and

t_d: development time in years.

2.2 COCOMO

The COnstructive COst MOdel (COCOMO) was developed by Barry Boehm [Boehm 81] [Boehm et al. 97]. COCOMO is a well-documented software costing model that can help predict the effort and duration of software projects. Three level of COCOMO models exists: basic, intermediate, detailed. Each level supports three classes or modes of software projects. The first class is organic, where relatively small teams work in a familiar environment and develop a well-understood application. The second mode is called semi-detached, this mode lies between the organic and embedded mode projects. The third one is the embedded class, where the project must work under such constraints as complex hardware or software.

The basic COCOMO model for predicting effort is the following formula which uses delivered source instructions (DSI) consisting of all lines of source code that exclude comments.

$$\text{Effort} = A(\text{KDSI})^b$$

where Effort = number of person-months to develop a project, KDSI = thousands of delivered source instructions (DSI), and A, b = constants values that depend on the mode of development (Organic: A=2.4, b=1.05, Embedded: A=3.6, b=1.20, and Semi-detached: A=3.0, b=1.12).

In 1995, Barry Boehm and his associates announced the release of COCOMO version 2, or COCOMO II. One of the objectives of COCOMO II was to develop a software cost and schedule estimation model of the 1990's and 2000's.

The COCOMO II model considers the following factors in attempting to produce the effort expressed in person months:

1. Precedentedness
2. Development Flexibility
3. Architecture/Risk Resolution
4. Team Cohesion
5. Process Maturity

Once a person-month value is determined, it is adjusted by seven cost drivers. These cost drivers are determined based on the following considerations:

1. Personnel capability
2. Product reliability and complexity
3. Required reusability
4. Platform difficulty
5. Personnel experience
6. Facilities
7. Schedule

The adjusted person-month figure can then be used to estimate cost and schedule.

2.3 Function Points

The function points metric was developed by Allen Albrecht [Albrecht and Gaffney 83]. The function points metric is intended to measure the "amount" of a software system as described by a requirement specification document.

There are two steps in calculating the function points metric. The first step is to count the user functions that consist of external inputs, external outputs, logical internal files, external interface files, and external inquiries. Each function has to be classified as simple, average, or complex, and assigned weights as shown in Table I.

TABLE I. Unadjusted Function Points (source: [Fenton 91])

ITEM	WEIGHTING FACTOR		
	SIMPLE	AVERAGE	COMPLEX
External input	3	4	6
External output	4	5	7
External inquiry	3	4	6
External interface file	5	7	10
Logical internal file	7	10	15

The total from this table is the number of unadjusted function points as calculated below.

$$\text{UFP} = \sum_{i=1}^{5}\sum_{j=1}^{3}(weight_{ij})*(count_{ij})$$

where UFP = Unadjusted Function Points.

The second step is to sum all the processing complexity factors. Albrecht has a list of 14 general system characteristics as shown [Fenton 91] in Figure 1, that are to be rated on a scale from 0 (no influence) to 5 (strong influence).

The fourteen general system characteristics are summarized into the GSC factor as follows. The sum of all general system characteristics is multiplied by 0.01 and added to 0.65 to obtain a weighting as shown below.

$$GSC = 0.65 + (0.01) \sum_{i=1}^{14} C_i$$

where GSC = general system characteristics, $0.65 \leq GSC \leq 1.35$, and C_i = complexity factors, $0 \leq C_i \leq 5$.

Factor contributing to complexity:	
F1 Reliable back-up and recovery	F2 Data communications
F3 Distributed function	F4 Performance
F5 Heavily used configuration	F6 On-line data entry
F7 Operation ease	F8 On-line update
F9 Complex interface	F10 Complex processing
F11 Reusability	F12 Installation ease
F13 Multiple sites	F14 Facilitate change

Figure 1. Albrecht's General System Characteristics (source: [Fenton 91])

The GSC factor is used to calculate the final function points measure as follows.

$$FP = UFP*GSC$$

where FP = Adjusted Function Points, UFP = Unadjusted Function Points, and GSC = general system characteristic.

The number of function points can be converted to lines of code for various languages. For example, the number of lines of code per function point for the C language is 128 [Jones 96a].

CHAPTER III

FUNCTION POINT CALCULATION

Allen Albrecht has been known as the inventor of function points since the method was first introduced in 1979 [Albrecht and Gaffney 83]. Albrecht was interested in productivity measurement in the software development process. The function points metric is intended to measure the amount of a software system as described by the user requirement specification, and also to measure software development and maintenance rates as well as sizes independently of the technology used for implementation [Jones 96].

The process of counting function points is simple enough to minimize the overhead of the measurement process, and it is also quite consistent as a measure among various projects and organizations. The validity of function points as a software size metric has been reinforced by the work of a number of researchers. These include Albrecht and Gaffney [Albrecht and Gaffney 83], Kemerer [Kemerer 87], Low and Jeffery [Low and Jeffery 90], and Kemerer and Porter [Kemerer and Porter 92].

Further, function points can measure the size of software relatively early in the development cycle and it is related to the user requirements in a manner easily understandable to the user [Garmus and Heron 96].

One of the early criticisms of the function point metric was that it was not suitable for use with other than management information or business systems. To address this issue, some adjustments of function points have been proposed such as function points MARK II, 3D function points, feature points, and object points. Other researchers also point to the success of the function points approach in a wide range of applications [Jones 96]. Jones recommends that the function point metric comes closer than any other measurement of software size.

In 1986, the non-profit International Function Points Users Groups (IFPUG) was formed to assist in transmitting data and information about the function point metric. In 1987, the British Government adopted a modified form of function points as the standard productivity metric. In 1994, IFPUG published release 4.0 of the Function Point Counting Practices Manual, which represented a consensus view of the rules for function points counting. This thesis refers to this manual for counting guidelines.

There are three steps in calculating the function point metric. The first step is to count the user functions that consist of external inputs, external outputs, logical internal files, external interface files, and external inquiries. Each function has to be classified as simple, average, or complex. The second step is to sum all the general system characteristics. Albrecht has a list of 14 general system characteristics that are to be rated on a scale from 0 (no influence) to 5 (strong influence). The third step is to convert the unadjusted function point factor to the adjusted function point factor.

3.1 Identify Counting Boundary

The first step of the function point counting procedure is to identify the counting boundary. The function point counting boundary indicates the border between the project or application being measured and the external applications [Garmus and Heron 96]. Boundaries are used to establish the scope of the product being measured, and are determined based on the user viewpoint.

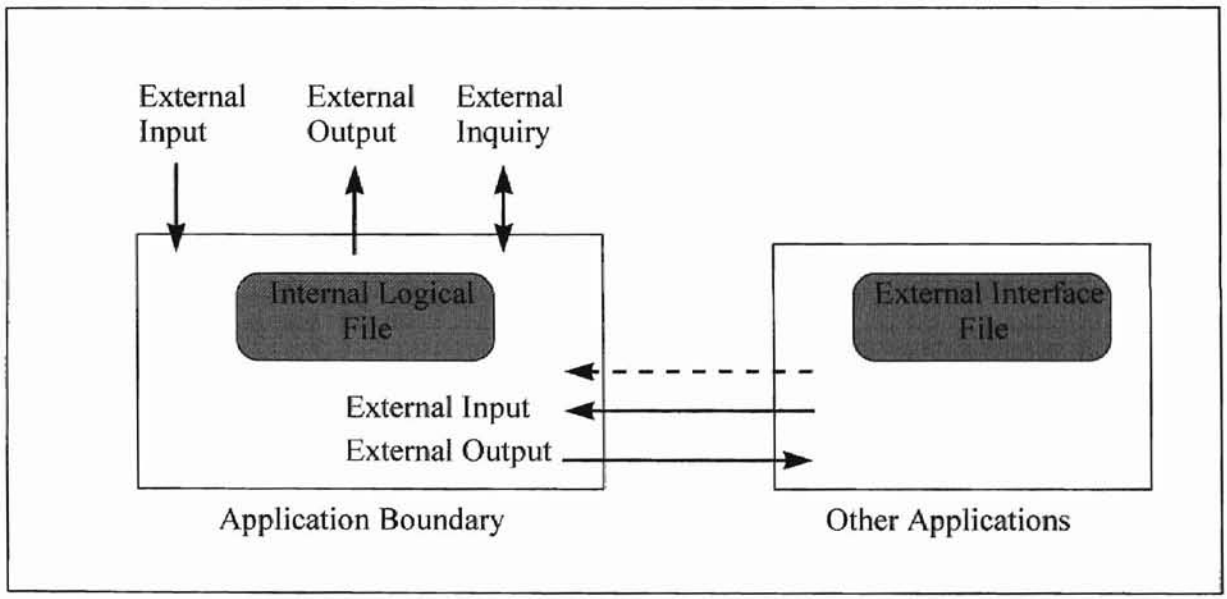


Figure 2. Function Point Counting Components (source: [Garmus and Heron 96])

Figure 2 displays the function point counting components. There may be more than one application included in the scope of a single project. If so, multiple boundaries would be identified and separately counted [Garmus and Heron 96].

3.2 Count Data Function Types

Data function types represent the functionality of data and are related to store, update, and retrieval. Data function types are defined as internal logical files (ILFs) and external interface files (EIFs). An internal logical file (ILF) is a user identifiable group of logically related data or control information maintained within the boundary of the application [Garmus and Heron 96].

An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application [Garmus and Heron 96]. This means an EIF counted for an application must be an ILF in another application.

The following paragraphs further explain the terms used in the definitions [Garmus and Heron 96].

Control Information: Control information is data used by the application to ensure compliance with business function requirements specified by the user.

User Identifiable: The term user identifiable, in the definitions of ILFs and EIFs, refers to the specific user requirements that an experienced user would define for the application.

Maintained: The term maintained, in the definitions of ILFs and EIFs, refers to the ability to modify data through an elementary process.

Elementary Process: An elementary process is the smallest unit of activity that is meaningful to the end user in the business.

The first step to count the data function types is to identify the data function types. The primary difference between an internal logical file and external interface file is that an internal logical file is maintained within the boundary of the application being counted, and an external interface file is read or referenced only within the boundary of the application being counted but maintained within a different application boundary.

TABLE II. Complexity Metric for Internal Logical Files and External Interface Files (source: [IFPUG 94])

	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	Low	Low	Average
2 to 5 RET	Low	Average	High
6 or more RET	Average	High	High

The next step is to assign the complexity of each internal logical file and external interface file based on the number of data element types (DETs), and record element types (RETs) as shown in Table II. Each of the terms are described below:

Data Element Type (DET) A data element type (DET) is a unique, user recognizable, and nonrecursive field in an ILF or EIF.

Record Element Type (RET) A record element type (RET) is a user recognizable subgroup of data elements within an ILF or EIF.

TABLE III. Unadjusted Function Point Weights for Internal Logical Files (source: [IFPUG 94])

Functional Complexity Rating	Unadjusted Function Points
Low	7
Average	10
High	15

TABLE IV. Unadjusted Function Point Weights for External Interface Files (source: [IFPUG 94])

Functional Complexity Rating	Unadjusted Function Points
Low	5
Average	7
High	10

The last step is to translate the number of internal logical files or external interface files to the number of unadjusted function points (UFP). Table III is used for translating the number of internal logical files to the number of unadjusted function points. Table IV is for translating the number of external interface files to the number of unadjusted function points.

3.3 Count Transaction Function Types

Transactional function types represent the functionality provided to the user for the processing of data by an application. Transactional function types are defined as external inputs (EIs), external outputs (EOs), and external inquiries (EQs).

3.3.1 External Inputs (EI)

An external input (EI) processes data or control information that comes from outside the application's boundary [Garmus and Heron 96].

The following paragraphs further explain the terms used within the definitions [Garmus and Heron 96] .

Control Information: Control information is data used by the application to ensure compliance with the business function requirements specified by the user.

Maintain: Maintain is the ability to modify or delete data through an elementary process.

Data: Data refers to the facts and/or figures processed by an input transaction.

The external input counting procedures have 3 steps. The first step is to identify the external inputs and the second is to determine the complexity of the external inputs. The last step is to translate the number of external inputs to unadjusted function points.

TABLE V. Complexity Metric for External Inputs (EI) (source: [IFPUG 94])

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	Low	Low	Average
2 FTR	Low	Average	High
3 or more FTR	Average	High	High

To identify external inputs, look for data or control information that fall within the definition of an external input and assign the complexity of an external input based on the number of file types referenced (FTRs) and the data element types (DETs). The complexity matrix for external inputs is presented in Table V. Each of the terms are described below [Garmus and Heron 96] :

File Type Referenced (FTR) A file type referenced is an internal logical file (ILF) or an external interface file (EIF) read or maintained by the external input.

Data Element Type (DET) A data element type is a unique, user recognizable, and nonrecursive field maintained on an internal logical file by the external input.

TABLE VI. Unadjusted Function Point Weights for External Inputs
(source: [IFPUG 94])

Functional Complexity Rating	Unadjusted Function Points
Low	3
Average	4
High	6

The last phase is the translation of the external inputs to unadjusted function points as shown in Table VI. For example, a high complexity rating translate to 6 unadjusted function points.

3.3.2 External Outputs (EO)

An external output (EO) is an elementary process that generates data or control information sent outside the application's boundary [Garmus and Heron 96] .

The following paragraphs further explain the terms used within the definitions [Garmus and Heron 96] .

Control Information: Control information is data used by an application to ensure compliance with business function requirements as specified by the user.

The external output counting procedure has three steps. The first step is to identify the external outputs and the second is to determine the complexity of the external outputs. The last step is to translate the number of external outputs to unadjusted function points.

TABLE VII. Complexity Metric for External Outputs (EO)
(source: [IFPUG 94])

	1 to 5 DET	6 to 19 DET	20 or more DET
0 to 1 FTR	Low	Low	Average
2 to 3 FTR	Low	Average	High
4 or more FTR	Average	High	High

To identify external outputs, look for data or control information that falls within the definition of an external output. The main identified rule is that the process sends data or control information outside the application's boundary.

TABLE VIII. Unadjusted Function Point Weights for External Outputs
(source: [IFPUG 94])

Functional Complexity Rating	Unadjusted Function Points
Low	4
Average	5
High	7

The next step is to assign the complexity of each external output based on the number of data element types (DETs) and file types referenced (FTRs) as shown in Table VII. The relevant terms are described below [Garmus and Heron 96] :

Data Element Type (DET) A data element type is a unique, user recognizable, and nonrecursive field that appears in the external output.

File Type Referenced (FTR) A file type referenced is a file read when the external output is processed. A file type referenced is counted for each internal logical file (ILF) or external interface file (EIF) read during the processing of the external output.

The last step consists of translating the number of external outputs to unadjusted function points as shown in Table VIII. For example, a high complexity rating translate to 7 unadjusted function points.

3.3.3 External Inquiries (EQ)

An external inquiry (EQ) is an elementary process made up of an input-output combination that results in data retrieval. No internal logical file (ILF) is maintained during processing [Garmus and Heron 96].

TABLE IX. Complexity Metric for External Inquiries (EQ)
(source: [IFPUG 94])

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	Low	Low	Average
2 FTR	Low	Average	High
3 or more FTR	Average	High	High

The following paragraphs further explain external inquiries by defining the terms used within the definition [Garmus and Heron 96] .

Maintain: Maintain is the ability to modify or delete data through an elementary process.

To identify external inquiries, identify where a request for data retrieval enters the application. IFPUG has defined specific rules that are used when identifing external inquiries. The following counting rules must apply before the information can be counted as an external inquiry:

1. An input request enters the application boundary.
2. Output results exit the application boundary.
3. Data is retrieved.

4. The process does not update an internal logical file.

The next step is to assign the complexity of each external inquiry based on the number of data element types (DETs) and file types referenced (FTRs), as shown in Table IX. The relevant terms are described below [Garmus and Heron 96] :

Data Element Type (DET) A data element type is a unique, user recognizable, and nonrecursive field that appears in the external inquiry.

File Type Referenced (FTR) A file type referenced is a file read when the external inquiry is processed. A file type referenced is counted for each internal logical file (ILF) or external interface file (EIF) read during the processing of the external inquiry.

TABLE X. Unadjusted Function Point Weights for External Inquiries (source: [IFPUG 94])

Functional Complexity Rating	Unadjusted Function Points
Low	3
Average	4
High	6

The last step is to translate the number of external inquiries to unadjusted function points as shown in Table X. For example, a high complexity rating translates to 6 unadjusted function points.

3.4 Determine Value Adjustment Factors

The value adjustment factor (VAF) is based on 14 general system characteristics (GSCs) that rate the general functionality of the application being counted. Each characteristic has an associated description that helps to determine the degrees of influence of the characteristics. The degrees of influence ranges on a scale of zero to five, from no influence to strong influence.

The 14 general system characteristics are summarized into the value adjustment factor. When applied, the value adjustment factor adjusts the unadjusted function point count by +/- 35 percent to produce the final function point count. The following is the formula to calculate the value adjustment factor.

$$\text{GSC} = 0.65 + (0.01) \sum_{i=1}^{14} C_i$$

where GSC = General System Characteristics, $0.65 \leq \text{GSC} \leq 1.35$, and C_i = complexity factors, $0 \leq C_i \leq 5$ (also mentioned in Section 3.5).

3.5 General System Characteristics

This section describes the International Function Point User Group (IFPUG) definitions, rules, and guidelines for identifying the value adjustment factor (VAF), which is used as a multiplier of the unadjusted function point count in order to calculate the final adjusted function point count of an application. The value adjustment factor (VAF) is

calculated based on the identification of fourteen general system characteristics. There characteristics are listed below.

1. Data Communications
2. Distributed Data Processing
3. Performance
4. Heavily Used Configuration
5. Transaction Rate
6. On-line Data Entry
7. End-user Efficiency
8. On-line update
9. Complex Processing
10. Reusability
11. Installation Ease
12. Operational Ease
13. Multiple Sites
14. Facilitate Change

Each general system characteristic (GSC) must be evaluated in terms of its degree of influence (DI) on a scale of zero to five:

0 Not present, or no influence

1 Incidental influence

2 Moderate influence

3 Average influence

4 Significant influence

5 Strong influence throughout

What follows is a brief description of the fourteen general system characteristics.

Data Communications

The data and control information used in the application are sent or received over communication facilities. The degree of influence for data communications is shown in Table XI.

TABLE XI. Degree of Influence for Data Communications
(source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	Application is pure batch processing or a standalone PC.
1	Application is batch but has remote data entry *or* remote printing.
2	Application is batch but has remote data entry *and* remote printing.
3	Application includes online data collection or TP (teleprocessing) front end to a batch process or query system.
4	Application is more than a front-end, but supports only one type of TP communications protocol.
5	Application is more than a front-end, and supports more than one type of TP communications protocol.

Distributed Data Processing

Distributed data or processing functions are a characteristic of the application within the application boundary. The degree of influence for distributed data processing is shown in Table XII.

TABLE XII. Degree of Influence for Distributed Data Processing
(source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	Application does not aid the transfer of data or processing function between components of the system.
1	Application prepares data for end user processing on another component of the system such as PC spreadsheets and PC DBMS.
2	Data is prepared for transfer, then is transferred and processed on another component of the system (not for end-user processing).
3	Distributed processing and data transfer are online and in one direction only.
4	Distributed processing and data transfer are online and in both directions.
5	Processing functions are dynamically performed on the most appropriate component of the system.

Performance

Application performance influences the design, development, installation, and support of the application. The degree of influence for performance is shown in Table XIII.

TABLE XIII. Degree of Influence for Performance
(source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No special performance requirements were stated by the user.
1	Performance and design requirements were stated and reviewed but no special actions were required.
2	Response time or throughput is critical during peak hours. No special design for CPU utilization was required. Processing deadline is for the next business day.
3	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
4	In addition, stated user performance requirements are stringent enough to require performance analysis tasks in the design phase.
5	In addition, performance analysis tools were used in the design, development, and/or implementation phases to meet the stated user performance requirements.

Heavily Used Configuration

A heavily used operational configuration, requiring special design considerations, is a characteristic of the application. The degree of influence for heavily used configuration is shown in Table XIV.

TABLE XIV. Degree of Influence for Heavily Used Configuration
(source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No explicit or implicit operational restrictions are included.
1	Operational restrictions do exist, but are less restrictive than a typical application. No special effort is needed to meet the restrictions.
2	Some security or timing considerations are included.
3	Specific processor requirements for a specific piece of the application are included.
4	Stated operation restrictions require special constraints on the application in the central processor or a dedicated processor.
5	In addition, there are special constraints on the application in the distributed components of the system.

Transaction Rate

The transaction rate is high and it influences the design, development, installation, and support of the application. The degree of influence for the transaction rate is shown in Table XV.

TABLE XV. Degree of Influence for Transaction Rate (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No peak transaction period is anticipated.
1	Peak transaction period (e.g., monthly, quarterly, seasonally, or annually) is anticipated.
2	Weekly peak transaction period is anticipated.
3	Daily peak transaction period is anticipated.
4	High transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks in the design phase.
5	High transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks and, in addition, require the use of performance analysis tools in the design, development, and/or installation phases.

On-line Data Entry

On-line data entry and control functions are provided in the application. The degree of influence for online data entry is shown in Table XVI.

TABLE XVI. Degree of Influence for On-line Data Entry (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	All transactions are processed in batch mode.
1	1% to 7% of transactions are interactive data entry.
2	8% to 15% of transactions are interactive data entry.
3	16% to 23% of transactions are interactive data entry.
4	24% to 30% of transactions are interactive data entry.
5	More than 30% of transactions are interactive data entry.

TABLE XV. Degree of Influence for Transaction Rate (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No peak transaction period is anticipated.
1	Peak transaction period (e.g., monthly, quarterly, seasonally, or annually) is anticipated.
2	Weekly peak transaction period is anticipated.
3	Daily peak transaction period is anticipated.
4	High transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks in the design phase.
5	High transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks and, in addition, require the use of performance analysis tools in the design, development, and/or installation phases.

On-line Data Entry

On-line data entry and control functions are provided in the application. The degree of influence for online data entry is shown in Table XVI.

TABLE XVI. Degree of Influence for On-line Data Entry (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	All transactions are processed in batch mode.
1	1% to 7% of transactions are interactive data entry.
2	8% to 15% of transactions are interactive data entry.
3	16% to 23% of transactions are interactive data entry.
4	24% to 30% of transactions are interactive data entry.
5	More than 30% of transactions are interactive data entry.

End-User Efficiency

The applications provided emphasize a design for end-user efficiency. The degree of influence for end-user efficiency is shown in Table XVII. The design includes the following.

- Navigational aids (for example, function keys, jumps, dynamically generated menus)
- Menus
- Online help and documents
- Automated cursor movement
- Scrolling
- Remote printing (via online transactions)
- Preassigned function keys
- Batch jobs submitted from online transactions
- Cursor selection of screen data
- Heavy use of reverse video, highlighting, colors underlining, and other indicators
- Hard copy user documentation of online transactions
- Mouse interface
- Pop-up windows
- As few screens as possible to accomplish a business function
- Bilingual support (if it supports two languages, count as four items)

TABLE XVII. Degree of Influence for End-user Efficiency (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	One to three of the above.
2	Four to five of the above.
3	Six or more of the above, but there are no specific user requirements related to efficiency.
4	Six or more of the above, and stated requirements for end-user efficiency are strong enough to require design tasks for human factors to be included (for example, minimize key strokes, maximize defaults, use of templates).
5	Six or more of the above, and stated requirements for end-user efficiency are strong enough to require use of special tools and processes to demonstrate that the objectives have been achieved.

On-line Update

The application provides online update for the internal logical files. The degree of influence for online update is shown in Table XVIII.

TABLE XVIII. Degree of Influence for On-line Update (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	None.
1	Online update of one to three control files is included. Volume of updating is low and recovery is easy.
2	Online update of four or more control files is included. Volume of updating is low and recovery easy.
3	Online update of major internal logical files is included.
4	In addition, protection against data lost is essential and has been specially designed and programmed in the system.
5	In addition, high volumes bring cost considerations into the recovery process. Highly automated recovery procedures with minimum operator intervention are included.

Complex Processing

Complex processing is a characteristic of an application. The degree of influence for complex processing is shown in Table XIX. The following are its components.

- Sensitive control (for example, special audit processing) and/or application specific security processing.
- Extensive logical processing.
- Extensive mathematical processing.
- Much exception processing resulting in incomplete transactions that must be processed again, for example, incomplete ATM transactions caused by TP interruption, missing data values, or failed edits.
- Complex processing to handle multiple input/output possibilities, for example, multimedia, or device independence.

TABLE XIX. Degree of Influence for Complex Processing
(source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	Any one of the above.
2	Any two of the above.
3	Any three of the above.
4	Any four of the above.
5	All five of the above.

Reusability

The application and the code in the application have been specifically designed, developed, and supported to be usable in other applications. The degree of influence for reusability is shown in Table XX.

TABLE XX. Degree of Influence for Reusability (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No reusable code.
1	Reusable code is used within the application.
2	Less than 10% of the application considered more than one user's needs.
3	Ten percent (10%) or more of the application considered more than one user's needs.
4	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
5	The application was specifically packaged and/or documented to ease re-use, and the application is customized for use by means of user parameter maintenance.

Installation Ease

Conversion and installation ease are characteristics of an application. The degree of influence for installation ease is shown in Table XXI.

TABLE XXI. Degree of Influence for Installation Ease (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No special considerations were stated by the user, and no special setup is required for installation.
1	No special considerations were stated by the user but special setup is required for installation.
2	Conversion and installation requirements were stated by the user, and conversion and installation guides were provided and tested. The impact of conversion on the project is not considered to be important.
3	Conversion and installation requirements were stated by the user, and conversion and installation guides were provided and tested. The impact of conversion on the project is considered to be important.
4	In addition to the two above, automated conversion and installation tools were provided and tested.
5	In addition to the three above, automated conversion and installation tools were provided and tested.

Operational Ease

Operational ease is a characteristic of an application. The degree of influence for operational ease is shown in Table XXII.

TABLE XXII. Degree of Influence for Operational Ease (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	No special operational considerations other than the normal back-up procedures were stated by the user.
1 - 4	One, some, or all of the following items apply to the application. Select all that apply. Each item has a point value of one, except as noted otherwise.
	Effective start-up, back-up, and recovery processes were provided, but operator intervention is required.
	Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items).
	The application minimizes the need for tape mounts.
	The application minimizes the need for paper handling.
5	The application is designed for unattended operation. Unattended operation means *no* operator intervention is required to operate the system other than to start up or shut down the application. Automatic error recovery is a feature of the application.

Multiple Sites

The application has been specifically designed, developed, and supported to be installed at multiple sites for multiple organizations. The degree of influence for multiple sites is shown in Table XXIII.

TABLE XXIII. Degree of Influence for Multiple Sites (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	User requirements do not require considering the needs of more than one user/installation site.
1	Needs of multiple sites were considered in the design, and the application is designed to operate only under identical hardware and software environments.
2	Needs of multiple sites were considered in the design, and the application is designed to operate only under similar hardware and/or software environments.
3	Needs of multiple sites were considered in the design, and the application is designed to operate under different hardware and/or software environments.
4	Documentation and support plan are provided and tested to support the application at multiple sites and the application is as described by 1 or 2.
5	Documentation and support plan are provided and tested to support the application at multiple sites and the application is as described by 3.

Facilitate Change

The application has been specifically designed, developed, and supported to facilitate change. The degree of influence for facilitate change is shown in Table XXIV. The following characteristics can apply for the application:

- Flexible query and report facility is provided that can handle simple requests.

- Flexible query and report facility is provided that can handle requests of average complexity.
- Flexible query and report facility is provided that can handle complex requests.
- Business control data is kept in tables that are maintained by the user with online interactive processes, but changes take effect only on the next business day.
- Business control data is kept in tables that are maintained by the user with online interactive processes, and the changes take effect immediately.

TABLE XXIV. Degree of Influence for Facilitate Change (source: [IFPUG 94])

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	Any one of the above.
2	Any two of the above.
3	Any three of the above.
4	Any four of the above.
5	All five of the above.

3.6 Adjusted Function Point Count

The last step to calculate function points is to calculating the adjusted function points. The following formula is for calculate adjusted function point counts.

$$FP = UFP*GSC$$

where FP = Adjusted Function Points, UFP = Unadjusted Function Points, and GSC = general system characteristics.

CHAPTER IV

FUNCTION POINT SENSITIVITY ANALYSIS

This section describes the experiment carried out to determine the sensitivity of the parameters involved in the calculation of the function point metric. The framework employed consisted of defining the purpose of the experiment, planning according to the available resources, tools that aided in the experiment, conducting the experiment, collecting data, and analyzing the results.

The aim of this experiment was to investigate the sensitivity of function point metric parameters, which are external inputs, external outputs, internal logical files, external interface files, and external inquiries. Sensitivity in this context means the impact of changes in the parameters on the function point metric.

4.1 Experimental Methodology

The methodology followed in this thesis is outlined below.

1. The first step takes as input the number of lines of code form the test programs.
2. The second step converts the number of lines of code to function points.

3. The third step calculates function points from the application documentation base on International Function Point User's Group (IFPUG) Counting Practice Manual, Version 4.0 [IFPUG 94].
4. The last step analyzes the sensitivity of the parameters (i.e., external inputs, external outputs, internal logical files, external interface files, and external inquiries), involved in function point calculation.

4.2 Software Tools Used

4.2.1 CSIZE

The tool used to measure the number of lines of code (LOC) of the test programs was CSIZE developed by Christopher Lott. It is available at the archives of the usenet newsgroup comp.software-eng (ftp.qucis.queensu.ca) [Malladi 96]. It gives as output the total number of lines of code, blank lines, lines with comments, non-blank lines, and non-comment lines, and preprocessor directives of C source code.

4.2.2 COSMOS

COSMOS, a software cost modeling system, is a freeware developed by the Department of Computer and Information Science at East Tennessee State University [Henry 97]. COSMOS is a tool for predicting the size of a software system and the amount of effort required to develop it. COSMOS is based on two classic models, Function Point Analysis and the Constructive Cost Model (COCOMO).

4.2.3 Microsoft Excel 97

Microsoft Excel 97 was used to analyze the data by plotting graphs. A graph can help to determine the sensitivity of the parameters (i.e., external inputs, external outputs, internal logical files, external interface files, and external inquiries), whether the change of input will change the slope of the curve.

4.3 Software Project Characteristics

The experimental software projects used in this thesis were from programs available on the Sequent Symmetry S/81 machine in the Computer Science Department at Oklahoma State University. The '/contrib' and '/usr/local' directories contain a collection of standard and non-standard programs, utilities, and games. These program vary in size from a few hundred lines of code to thousands of lines of code, and represent various fields of computer science. The complete listing of all the programs is included in Appendix B.

4.4 Function Points and Line of Code

There exists a strong correlation between function points and lines of code. The most popular conversion factor table is from Capers Jones [Jones 96]. The conversion factor for the COBOL language is 145 lines of code per function point. For the C language, the conversion factor is 128 lines of code per function point [Jones 96].

4.5 Sensitivity Analysis of Function Point

There are three steps in function point calculation. The first is counting the five parameters which are internal logical files, external interface files, external inputs, external outputs, and external inquiries. The second step is determining the fourteen general system characteristics. The third step is that converting the unadjusted function point count to adjusted function point count.

From the methodology of this thesis, first we count the number of lines of code, and then convert it to the number of function points. For example, Table XXV shows that the XGOPHER software has 11769 lines of code which is approximately equal to 100 function points.

TABLE XXV. Number of Lines of Code Converted to Number of Function Points

Software Project	Language	Lines of Code (LOC)	Function Points
1. XGOPHER	C	11769	100
2. UTREE	C	9413	70
3. GHOSTVIEW	C	8475	65
4. SC	C	8343	63
5. GZIP	C	5220	40

An example of XGOPHER software using the COSMOS software tool is presented in Figure 3. The XGOPHER software measured in this experiment contains 16 external inputs, 16 external outputs, 57 logical internal files, 17 external interface file,

and 16 external inquiries, for a total of 122 unadjusted function points (UFP) when all functions have an average weight classification.

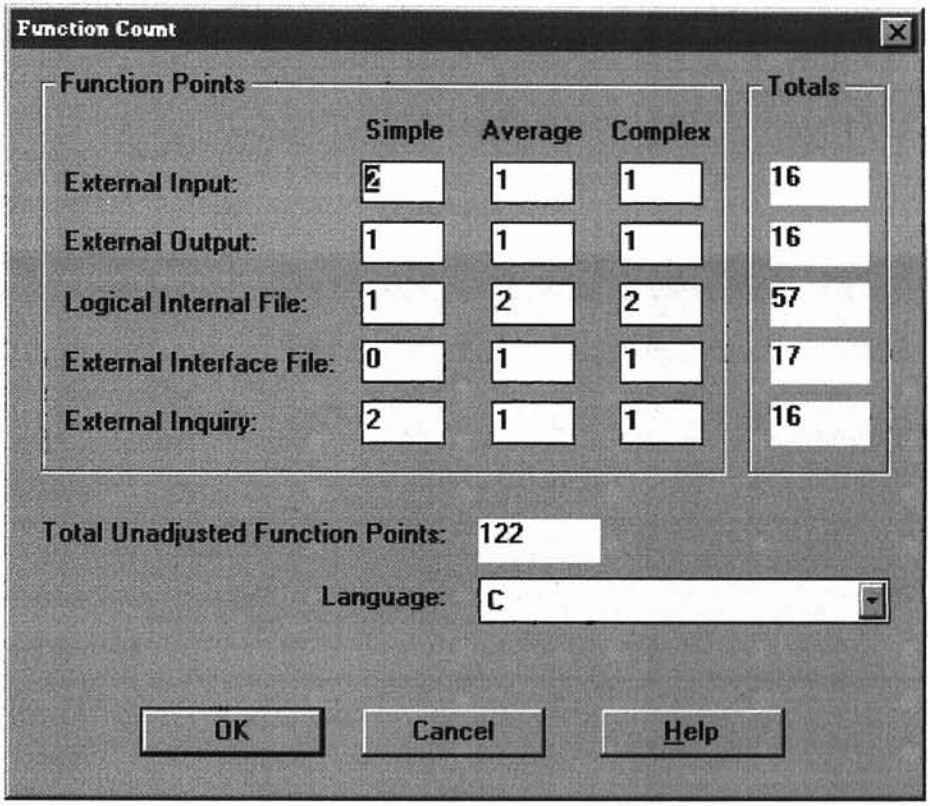


Figure 3. COSMOS Function Count Dialog Box of XGOPHER Program

The function point counting process we just completed generates the unadjusted function point count (Figure 3). The next step is to adjust the number of unadjusted function points. The general system characteristics measure the degree of influence for each of the fourteen categories. The total of the general system characteristics for XGOPHER is calculated as 0.17. The general system characteristics of online data entry, online update, complex processing, operational ease, multiple sites, and facilitate change

have no influence and thus their value is equal to zero. The heavily used configuration and installation ease were rated as insignificant, resulting in a value of one. The data communications, distributed functions, end user efficiency, and reuseability were rated as moderate, resulting in a value of two. The performance was rated as average with a value of three. The transaction rate was rated as significant with a value equal to four.

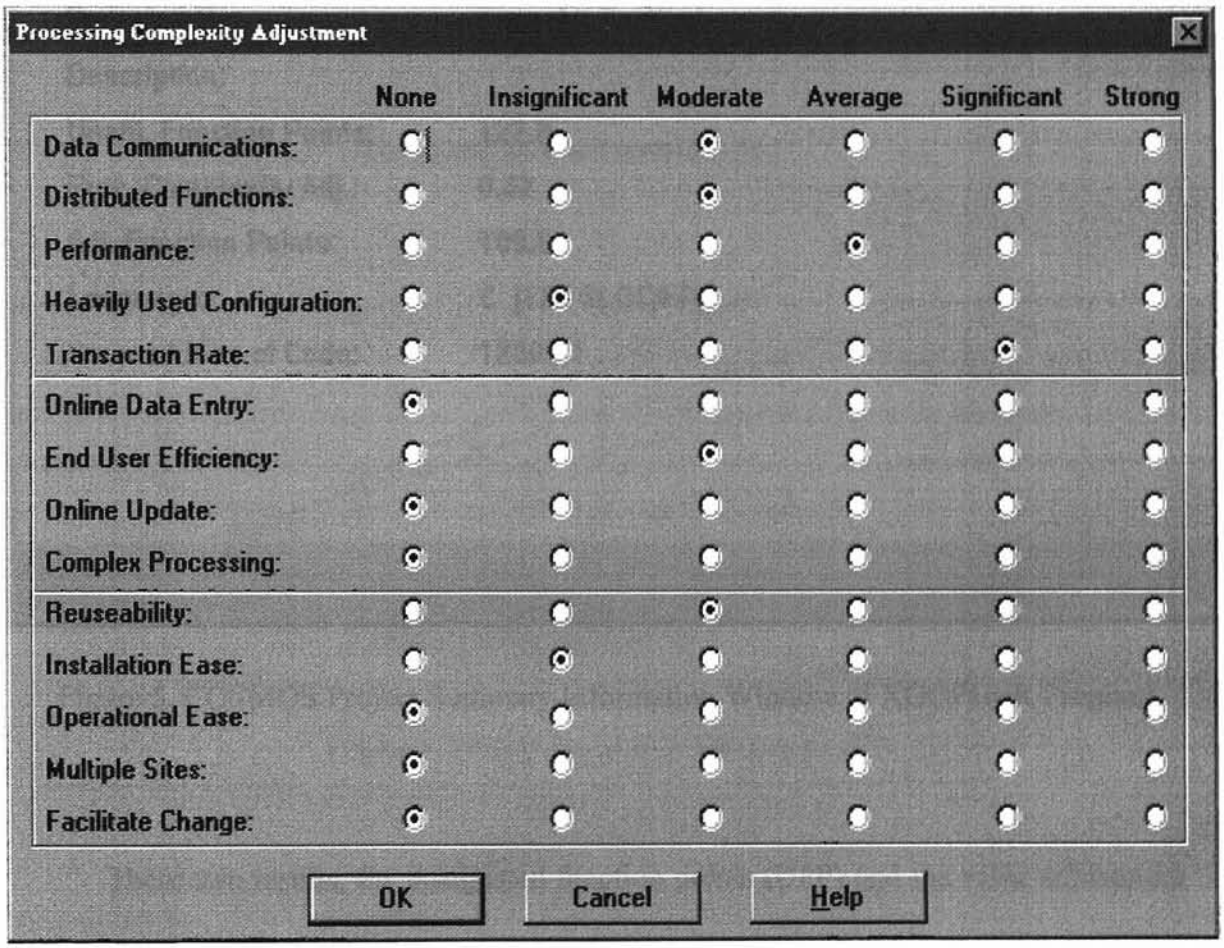


Figure 4. COSMOS Complexity Adjustment Dialog Box of XGOPHER Program

The general system characteristics of 0.17 is then added to 0.65 to get the value adjustment factor (VAF) which falls within the adjustment range of +/- 35 %.

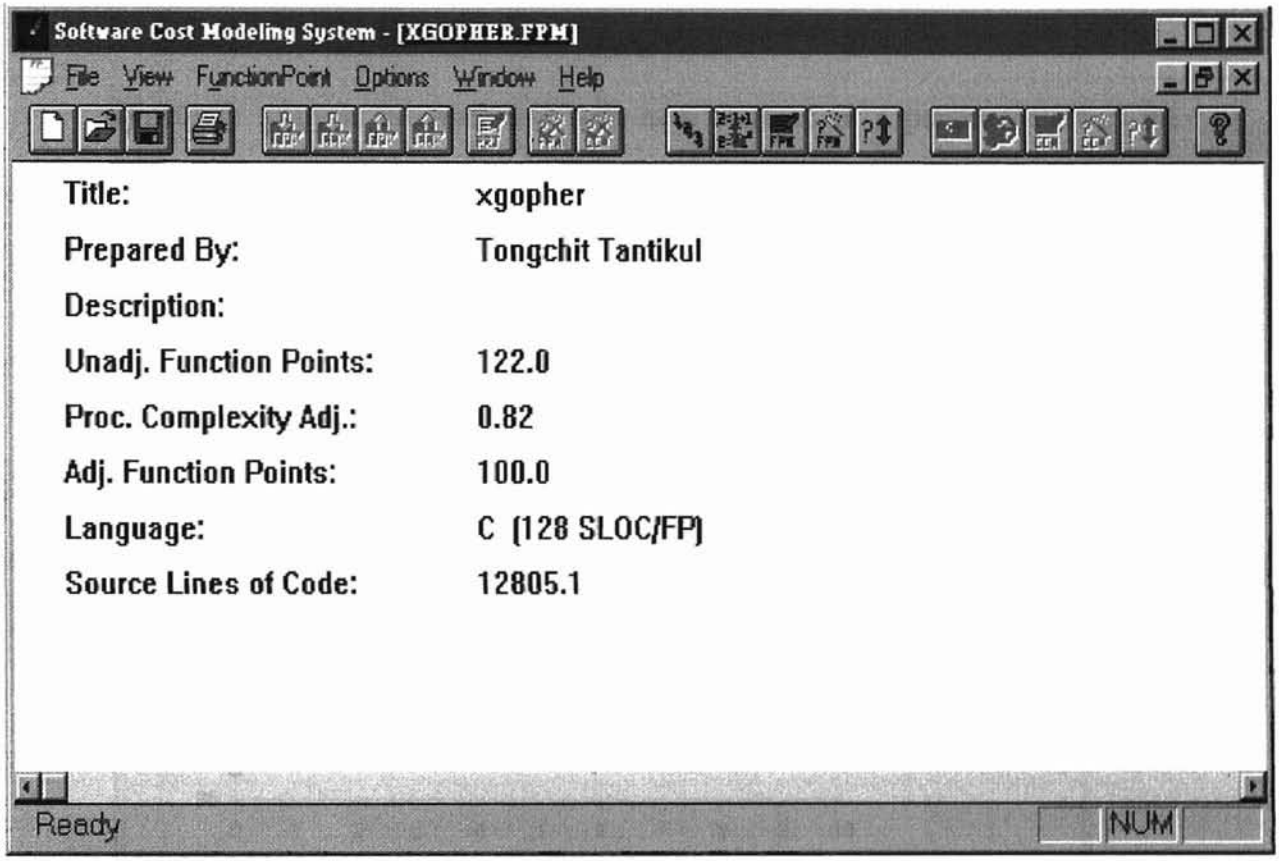


Figure 5. COSMOS Project Summary Information Window of XGOPHER Program

These two results, the unadjusted function points (UFP) and the value adjustment factor (VAF), are then combined to produce 100 as the adjusted function points as shown in Figure 5.

In the following subsections, five separate analyses are performed. Each analysis determines the sensitivity among the function types complexity that are simple, average, and complex for each parameter.

4.5.1 Sensitivity Analysis of External Inputs (EI)

Sensitivity analysis is concerned with how changes in a parameter affect the optimal solution.

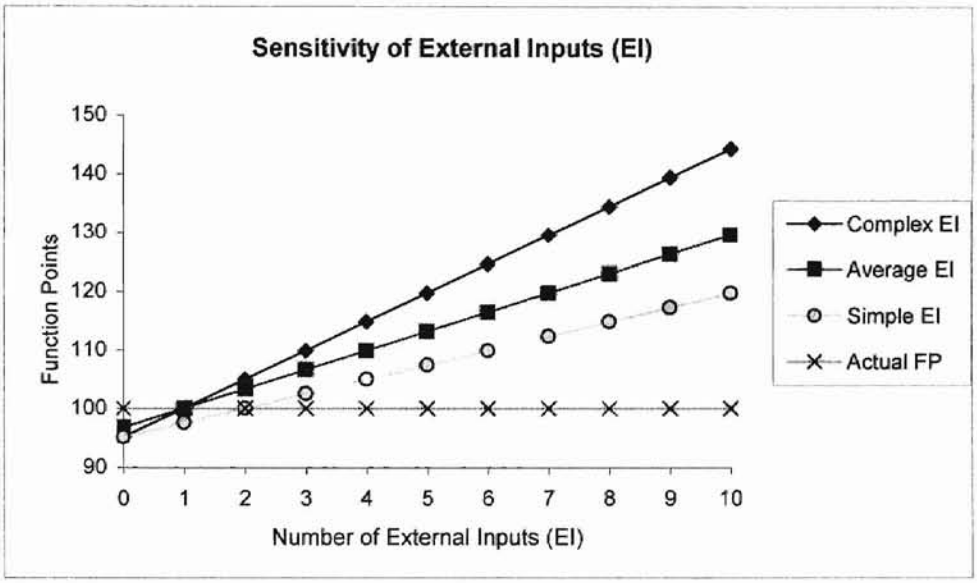


Figure 6. Sensitivity of External Inputs (EI)

Figure 6 shows the resulting sensitivity curve of he number of external inputs of the XGOPHER program. In this situation, the optimal solution is equal to 100, the complex external input is 1, the average external input is 1, and the simple external inputs are 2.

So the total of external inputs is 16:

External input (EI) = 3*(Simple EI) +4*(Average EI)+ 6*(Complex EI)

= 3*(2) + 4*(1) + 6*(1)

= 16

An increase in the number of complex external inputs will result in the highest difference from the actual function points. So the complex external input is more sensitive than the average and simple external input.

4.5.2 Sensitivity Analysis of External Outputs (EO)

The sensitivity of the number of external outputs is shown in Figure 7. Figure 7 is for the example of the XGOPHER program. The optimal value of simple external output is 1, average external output is 1, and complex optimal output is also 1.

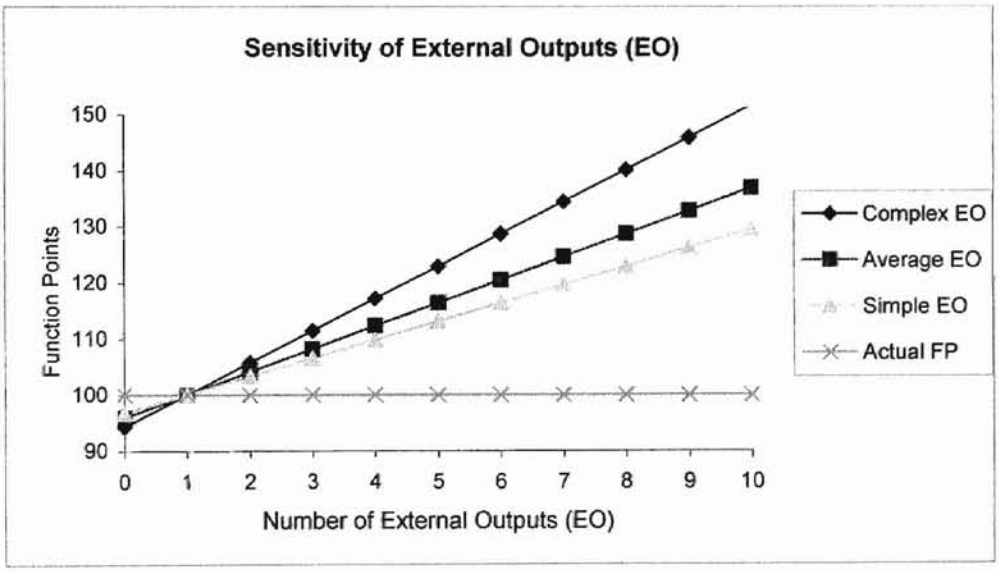


Figure 7. Sensitivity of External Outputs (EO)

The total of external output is 16:

External output (EO) = 4*(Simple EO) +5*(Average EO)+ 7*(Complex EO)

= 4*(1) + 5*(1) + 7*(1)

= 16

Form the curve in Figure 7, the complex external output is very sensitive.

4.5.3 Sensitivity Analysis of Internal Logical Files

The sensitivity of the number of internal logical files (ILF) to the number of function points is shown in Figure 8.

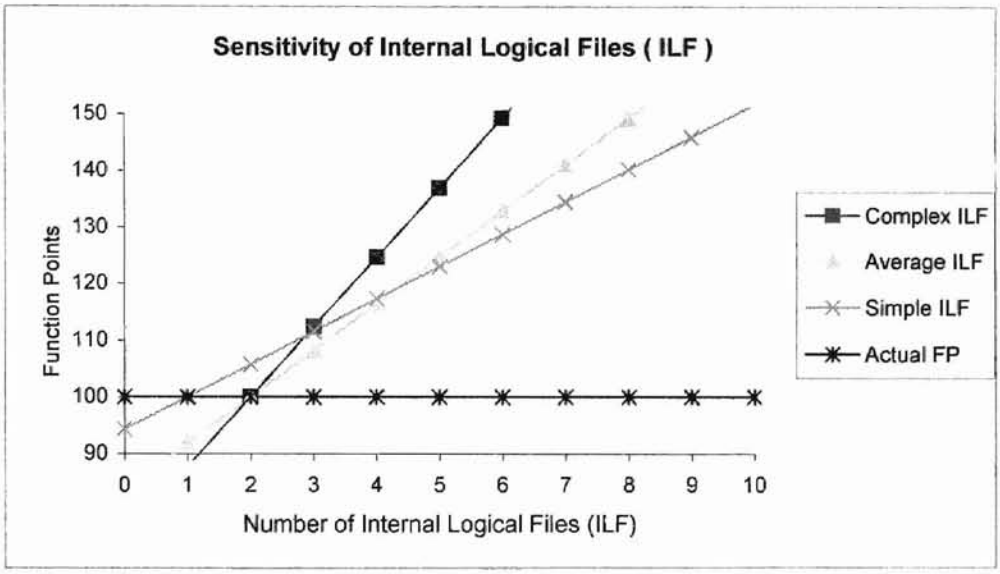


Figure 8. Sensitivity of Internal Logical Files (ILF)

Figure 8 is for the case of XGOPHER program. The optimal values for simple internal logical files, average internal logical files, and complex internal logical files are 1, 2, and 2, respectively. So the total of internal logical files is 57:

Internal logical file (ILF) = 7*(Simple ILF) +10*(Average ILF)+ 15*(Complex ILF)

= 7*(1) + 10*(2) + 15*(2)

= 57

From the curve in Figure 8, the number of complex internal logical files is most sensitive.

4.5.4 Sensitivity Analysis of External Interface Files

Figure 9 shows the resulting sensitivity curve of the number of external interface files of the XGOPHER program.

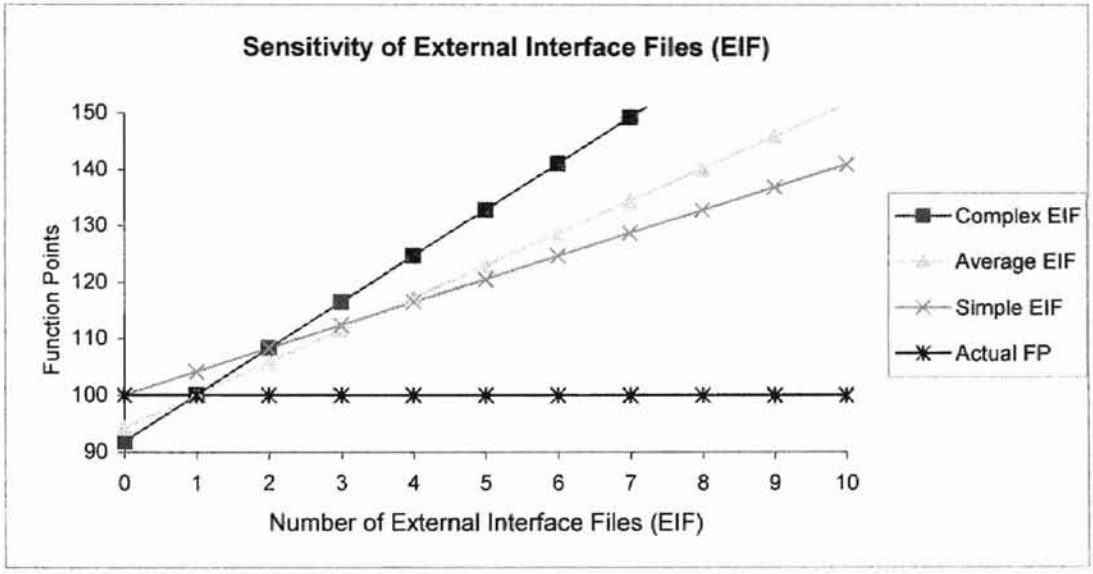


Figure 9. Sensitivity of External Interface Files (EIF)

The function point optimal solution of XGOPHER is equal to 100, when the complex external interface file is 1, the average external interface file is 1, and the simple external interface file is 0. So the total of external inputs is 17:

External interface file (EIF) = 5*(Simple EIF) +7*(Average EIF)+ 10*(Complex EIF)

$= 5*(0) + 7*(1) + 10*(1)$

$= 17$

An increase in the number of complex external interface files will result in the highest difference from the actual function points. So the complex external interface files is more sensitive than the average and simple external interface file.

4.5.5 Sensitivity Analysis of External Inquiries

The sensitivity of the number of external inquiries is shown in Figure 10 for the case of the XGOPHER program.

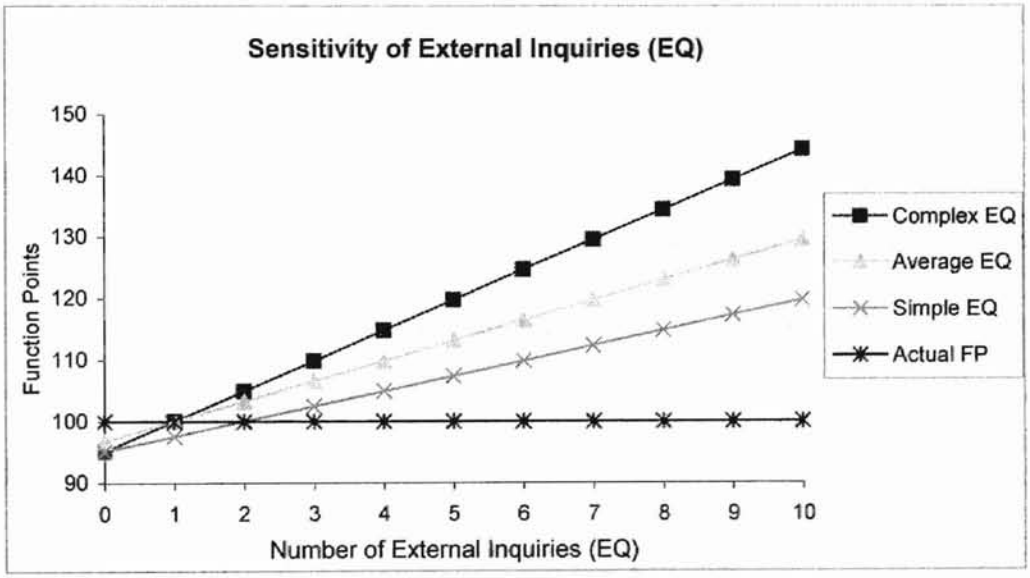


Figure 10. Sensitivity of External Inquiries (EQ)

The function point optimal value of the XGOPHER program is equal to 100, when the complex external inquiries is 1, the average external inquiries is 1, and the simple external inquiries is 2. So the total number of external inquiries is 16:

External inquiry (EQ) = 3*(Simple EQ) +4*(Average EQ)+ 6*(Complex EQ)

= 3*(2) + 4*(1) + 6*(1)

= 16

An increase in the complex external inquiries will result in the highest difference from the actual function points. So the number of complex external inquiries is more sensitive than the average and simple external inquiries.

4.5.6 Sensitivity of Five Parameters of Function Point Calculation

In the final step of the methodology to determine the sensitivity analysis of function points parameters, all the parameters (i.e., external inputs, external outputs, internal logical files, external interface files, and external inquiries) were plotted together [Burr and Owen 96].

Figure 11 shows the sensitivity of the function point parameters for the XGOPHER program. The complex external inputs (EI) is 1. The complex external outputs (EO) is 1. The complex external interface files (EIF) is 1. The complex internal logical files (ILF) is 2. The complex external inquiries (EQ) is 1. The optimal adjusted function points is 100. The optimal unadjusted function points (UFP) is 122.

$$\text{UFP} = \sum_{i=1}^{5}\sum_{j=1}^{3}(weight_{ij})*(count_{ij})$$

where UFP = Unadjusted Function Points.

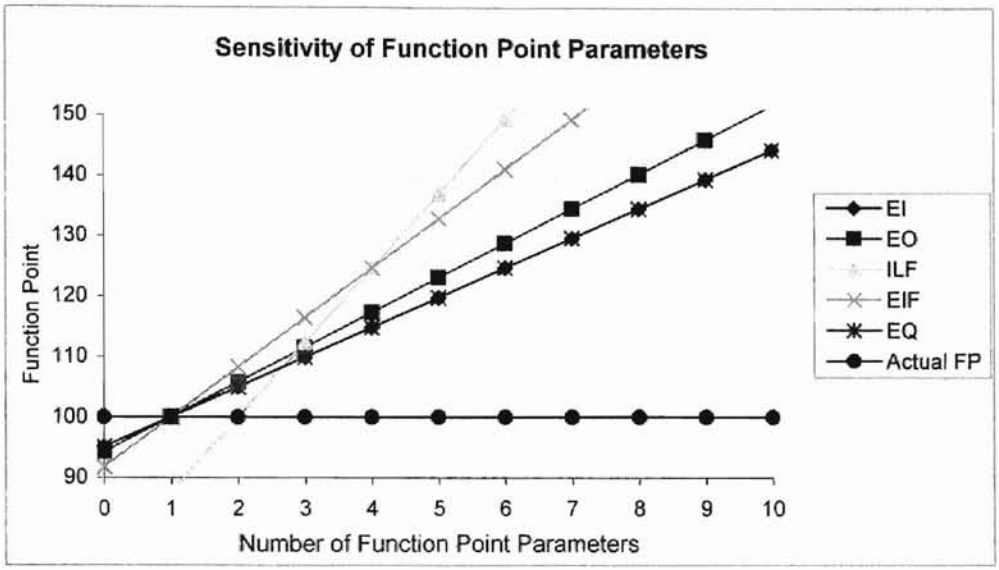


Figure 11. Sensitivity of Five Parameters for Function Point Calculation

The unadjusted function points can be expressed in the linear programming models as follows [Winston 94]:

(New optimal UFP value) = (Old optimal UFP value) + Δb_i

where UFP = Unadjusted Function Points,

b_i = weighting factor of each parameter

for external inputs (EI): UFP = 122 + 6Δ

for external outputs (EO): UFP = 122 + 7Δ

for external inquiries (EQ): UFP = 122 + 6Δ

for external interface files (EIF): UFP = 122 + 10Δ

for internal logical files (ILF): UFP = 122 + 15Δ

From the constraints we can see that a one unit increase in the number of external inputs (EI) will increase the optimal unadjusted function points (UFP) value by 6, external outputs (EO) by 7, external inquiries (EQ) by 6, external interface files (EIF) by 10, and internal logical files (ILF) by 15. Thus, the number of internal logical files (ILF) is the most sensitive parameter.

4.5.7 Sensitivity Analysis of General System Characteristics

This subsection explains the sensitivity of the fourteen general system characteristics.

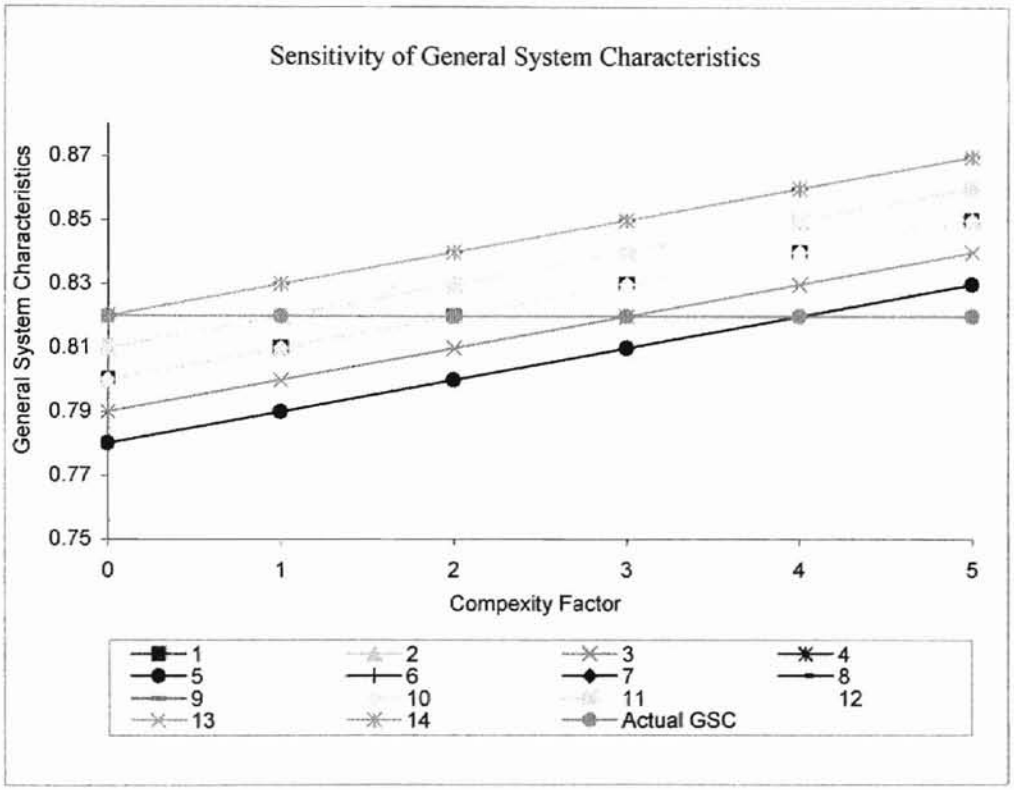


Figure 12. Sensitivity of General System Characteristics

The general system characteristics are rated on a scale from 0 (no influence) to 5 (strong influence). The sum of all general system characteristics is multiplied by 0.01 and added to 0.65 to obtain an adjustment factor:

$$GSC = 0.65 + (0.01) \sum_{i=1}^{14} C_i$$

where GSC = General System Characteristics , $0.65 \leq GSC \leq 1.35$, and C_i = complexity factors, $0 \leq C_i \leq 5$.

Figure 12 shows the sensitivity of the general system characteristics of the XGOPHER program. The optimal solution is 0.82. The slopes of the lines are equal, so no parameter is more sensitive than the others.

CHAPTER V

SUMMARY AND FUTURE WORK

5.1 Summary

In Chapter I, a general definition for and the importance of software metrics was introduced. The main goal of this thesis also was stated. Chapter II discussed the definition of software cost and complexity models such as the Constructive Cost Model (COCOMO), the Software Life Cycle Model (SLIM), and the Function Point Model. Chapter III of this thesis described the function point calculation in detail. Chapter IV described the experimental methodology and sensitivity analysis of function point calculation.

The main purpose of this thesis was to study the function point metric and analyze its sensitivity. The parameters involved in function point calculation (i.e., external inputs, external outputs, external inquiries, internal logical files, and external interface files) were studied with respect to their sensitivity on various programs in the test suites.

The results, even though they are based on a relatively small sample size of test programs, are reasonably representative. The results of the experiment on the sensitivity analysis of the function point metric are given in Appendices D and E as a series of plots between the number of lines of code and various function points parameters.

From the experiment, it was found that the number of internal logical files is the most sensitive parameter. Changes in the value of the number of internal logical files will affect the number of function points most. Adjusting the fourteen General System Characteristics (see Section 4.5.7) does not change the number of function points significantly. The fourteen General System Characteristics are not as sensitive to size changes as the number of internal logical files.

5.2 Future Work

Possible future work to extend and utilize the work done in this thesis includes the following: performing a study by using a larger number and a wider variety of test programs, experimenting with other programming languages such as C^{++}, Java, and Ada, and using the results of the sensitivity analysis to propose a number of modifications to the process of calculating function points.

REFERENCES

[Albrecht and Gaffney 83] A. J. Albrecht and J. E Gaffney, " Software Function, Source Line of Code, and Development Effort Prediction: a Software Science Validation," *IEEE Transaction on Software Engineering*, Vol. 9, No. 6, pp. 639-648, November 1983.

[Bennatan 95] E. M. Bennatan, *On Time, Within Budget: Software Project management Practice and Techniques*, McGraw-Hill Book, New York, NY, 1995.

[Boehm 81] B. W. Boehm, *Software Engineering Economics*, Prentice-Hall, New York, NY, 1981.

[Boehm et al. 97] B.W. Boehm, Christ Abts, Brad Clark, and Sunita Devnani-Chulani, "COCOMO II Model Definition Manual," COCOMOII Research Project, Available at: http://sunset.usc.edu/COCOMOII/Cocomo.html, October 1997.

[Burr and Owen 96] Adrian Burr and Mal Owen, *Statistical Methods for Software Quality: Using Metrics to Control Process and Product Quality*, International Thomson Computer Press, Boston, MA, 1996.

[Conte et al. 86] S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics and Models*, The Benjamin Cummings Publishing Company, Inc., Menlo Park, CA, 1986.

[Fenton 91] Norman E. Fenton, *Software Metrics: A Rigorous Approach*, Chapman&Hall, London, UK, 1991.

[Garmus and Heron 96] David Garmus and Davis Herron, *Measuring the Software Process: A Practical Guide to Functional Measurements,* Prentice Hall, Upper Saddle River, NJ, 1996.

[Henry 97] Joel Henry, "The Software Cost Modeling System (COSMOS)," COSMOS Research Project, Available at: http://csvaxsrv.etsu-tn.edu/~dsgnstudio/cosmos/, August 1997.

[IFPUG 94] International Function Point Users Group, *Counting Practices Manual*, Release 4.0, Westerville, OH, 1994.

[Jones 96a] Capers Jones, *Applied Software Measurement: Assuring Productivity and quality*, 2nd Edition, McGraw Hill, New York, NY 1996.

[Jones 96b] Capers Jones, *Software System Failure and Success,* International Thomson Computer Press, Boston, MA, 1996.

[Kemerer 87] Chris F. Kemerer, "An Empirical Validation of Software Cost Estimation Models," *Communications of the ACM*, Vol. 30, No. 5, pp. 416-429, May 1987.

[Kemerer 93] Chris F. Kemerer, "Reliability of Function Points Measurement: A Field Experiment," *Communications of the ACM*, Vol. 36, No. 2, pp. 85-97, February 1993.

[Kemerer and Porter 92] Chris F. Kemerer and Benjamin S. Porter, "Improving the Reliability of Function Point Measurement: An Empirical Study," *IEEE Transaction on Software Engineering*, Vol. 18, No. 11, pp. 1011-1024, November 1992.

[Kitchenham 96] Barbara Kitchenham, *Software Metrics: Measurement for Software Process Improvement*, Blackwell Publishers Inc., Cambridge, MA, 1996.

[Low and Jeffery 90] Graham C. Low and D. Ross Jeffery, "Function Points in the Estimation and Evaluation of the Software Process," *IEEE Transaction on Software Engineering*, Vol. 16, No. 1, pp. 64-71, January 1990.

[Malladi 96] Srikanth Reddy Malladi, *Investigation of the Relationship Between Volume Metric and Symbol Table*, MS Thesis, Computer Science Department, Oklahoma State University, Stillwater, OK, 1996.

[Matson et al.94] Jack E. Matson, Bruce E. Barrett, and Joseph M. Mellichamp, "Software Development Cost Estimation Using Function Points," *IEEE Transaction on Software Engineering*, Vol. 20, No. 4, pp. 275-287, April 1994.

[McDermid 94] John A. McDermid, *Software Engineer's Reference Book*, CRC Press, Inc., Boca Raton, FL, 1994.

[Pressman 97] Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 4th Edition, Mcgraw Hill, New York, NY 1997.

[Putnam 80] L. Putnam, *Software Cost Estimation and Life Cycle Control*, IEEE Computer Society Press, Los Alamitos, CA, 1980.

[Sommerville 96] Ian Sommerville, *Software Engineering*, 5th Edition, Addison-Wesley Publishing Company, New York, NY, 1996.

[Winston 94] Wayne L. Winston, *Operation Research: Applications and Algorithms*, International Thomson Publishing, Belmont, CA, 1994.

APPENDICES

APPENDIX A

GLOSSARY

Adjusted Function Point Count:
: The total function point count based on the unadjusted function point count multiplied by the value adjustment factor. The count is calculated using a specific formula for development project, enhancement project, and application. The adjusted count is commonly called the function point count.

Application:
: A cohesive collection of automated procedures and data supporting a business objective. It consists of one or more components, modules, or subsystems. Frequently used synonymously with System, Application System, and Information System.

Boundary:
: The border between the application or project being measured and the external applications or the user domain. A boundary establishes what functions are included in the function point count.

COCOMO
: The Constructive Cost Model, published originally in 1981 by Barry Boehm, which defines a method of estimating required software project effort and schedule.

Complex Processing:
: One of the 14 general system characteristics describing the degree to which processing logic influences the development of the application.

Control Information:
: Data used by the application to ensure compliance with business function requirements specified by the user.

Data Attribute:
: A characteristic of an entity. Attributes are generally analogous to data element types (DETs).

Data Communications: One of the 14 general system characteristics describing the degree to which the application communicates directly with the processor.

Data Element Type (DET): A unique, user recognizable, and nonrecursive field. The number of DETs is used to determine the complexity of each function type and the function type's contribution to the unadjusted function point count.

Data Function Types: The functionality provided to the user to meet internal and external data requirements. Data function types are either internal logical files (ILFs) or external interface files (EIFs).

Degree of Influence (DI): A numerical indicator of the amount of impact of each of the 14 general system characteristics, ranging from zero to five. These indicators are used to compute the value adjustment factor.

Distributed Processing: One of the 14 general system characteristics describing the degree to which the application transfers data among components of the application.

Development: The specification, construction, testing, and delivery of a new information system.

EIF: See External Interface File.

Elementary Process: The smallest unit of activity that is meaningful to the end user in the business. It must be self-contained and leave the business of the application being counted in a consistent state.

End-User Efficiency: One of the 14 general system characteristics describing the degree of consideration for human factors and ease of use for the user of the application measured.

External Input: A user data type or user control type that crosses the application boundary and adds, modifies, or deletes information from logical internal files.

External Inquiry: An input/output combination that crosses the application boundary and retrieves data or control information for immediate response. The inquiry does not modify data.

External Interface File: A set of logically related data used outside the application.

External Output: A user data type or user control type that leaves the application boundary.

Facilitate Change: One of the 14 general system characteristics describing the degree to which the application has been developed for easy modification of processing logic or data structure.

File: For data function types, a logically related group of data, not the physical implementation of those groups of data.

File Type Referenced (FTR): An ILF or EIF read or maintained by a transactional function type.

Function: The features or capabilities of an application as seen by the customer/user.

Function Point: A metric that describes a unit of work product suitable for quantifying application software.

Function Point Analysis: A standard method for measuring software development and maintenance from the customer's point of view.

Function Point Count: The function point measurement of a particular application or project.

Function Type: The five basic information services provided to the user of an application and identified in function point analysis. The five function types are external input, external output, external inquiry, internal logical file, and external interface file.

Functional Complexity: A specific function type's complexity rating as low, average, or high. For data function types, the complexity is determined by the number of RETs and DETs. For transactional function types, the complexity is determined by the number of FTRs and DETs.

General System Characteristics (GSCs): A set of 14 questions that evaluate the overall complexity of an application.

Heavily Used Configuration: One of the 14 general system characteristics describing the degree to which computer resource restrictions influenced the development of an application.

IFPUG:	The International Function Point Users Group, a non-profit organization.
ILF	See Internal Logical File.
Installation Ease:	One of the 14 general system characteristics describing the degree to which conversion from previous environments influenced the development of the application.
Internal Logical File:	One of the two data function types. An ILF is a user identifiable group of logically related data or control information maintained within the boundary of the application being counted. See also External interface file.
Maintained:	The ability to modify data through an elementary process.
Maintenance:	The effort to keep an application performing according to its specifications, generally without changing its functionality (or function point count). Maintenance includes repair, minor enhancement, conversion, user support and preventive maintenance activities. Activities include defect removal, hardware or software upgrades, optimization or quality improvement, and user support.
Multiple Sites:	One of the 14 general system characteristics describing the degree to which the application has been developed for multiple locations and user organizations.
Online Data Entry:	One of the 14 general system characteristics describing the degree to which data is entered through interactive transactions.
Online Update:	One of the 14 general system characteristics describing the degree to which internal logical files are updated online.
Operational Ease:	One of the 14 general system characteristics describing the degree to which the application attends to operational aspects, such as, start-up, back-up, and recovery processes.
Performance:	One of the 14 general system characteristics describing the degree to which response time and throughput performance considerations influenced the application development

Person Month: A unit of measurement for effort used in estimating schedules. This is the amount of work a single person can do in one calendar month.

Project: A collection of work tasks with a time frame and a work product to be delivered.

Record Element Type (RET): User recognizable subgroups of data elements within an ILF or EIF. There are two types of RETs : mandatory and optional.

Reusability: One of the 14 general system characteristics describing the degree to which the creation and/or use of standardized software specifications influenced the development of the application.

Source Lines of Code (SLOC): The number of lines of code delivered for a project, excluding comment lines.

Transaction Rate: One of the 14 general system characteristics describing the degree to which the rate of business transactions influenced the development of the application.

Transactional Function Type: The functionality provided to the user to process data by an application. Transactional function types are defined as external inputs, external outputs, and external inquiries.

Unadjusted Function Point Count (UFPC): The specific countable functionality provided to the user by the project or application. It has two function types—data and transactional.

User: (1) The person or organization that uses the measured application. Included would be the requirements author, end users, management users, auditors, and operations. (2) The human beings that use the application.

User Identifiable: Specific user requirements that an experienced user would define for the application.

User View: The application as seen or perceived by the user. It is the external logical view of the business functions being performed rather than the technical or processing view. There may be multiple user communities for the same application.

Value Adjustment Factor (VAF): The factor that indicates the general functionality provided to the user of the application. The VAF is calculated based on an assessment of the 14 general system characteristics (GSCs) for an application.

APPENDIX B

SOURCE AND BRIEF DESCRIPTON OF PROGRAMS IN THE TEST SUITE

PROGRAM FROM '/local/src' ON SEQUENT

PROGRAM	BRIEF DESCRIPTION
archie	a program that queries anonymous ftp databases
forwarder	a mail forwarding program
ghostview	a utility to view PostScript documents
man	a global man page extraction utility
rs	a reminder service program
sc	a spreadsheet calculator program
stat	a program that contains a collection of command level functions that can be interconnected to from a statistical network
utree	a screen oriented file system browser and utility
xcalender	a calendar program with a notebook for X
xgopher	a gopher client program for X window sytem
xnlock	a screen lock program
xpostit	a program for manipulating on-screen post-it notes

PROGRAM FROM '/contrib/src' ON SEQUENT

PROGRAM	BRIEF DESCRIPTION
agrep	an approximate regular expression matching utility
animate	a program that displays a sequence of images
ce	a simple unix text editor
combine	a program that combines images to create new image
convert	a program that converts images in one form to another
display	a program that displays an image on any workstation running X
fill	a simple text formatter program
fplan	a flight planner program
gzip	a utility to compress files
import	a program that captures an X server screen
indent	a program that inserts or deletes spaces in a file
lander	a game program
mogrify	a program that transforms an image or a sequence of images. These transforms include image scaling, image rotation, color reduction and others
montage	a program crestes a composite image by combining several separate images
othello	a game program
par	a filter program for reformatting paragraphs
sail	a game program
segment	a program that segments an image
uuconvert	a program that uudecodes a uuencoded file
xalarm	an alarm clock program for X
xasteroids	an X windows based game program
xdl	a program that displays DL animation
xgetftp	an X based tool for browsing and retrieving files from ftp sites
xtalk	a talk program between two X terminals

APPENDIX C

LINE OF CODE MEASURES OBTAINED BY USING THE CSIZE SOFTWARE TOOL [Malladi 96]

PROGRAM	LOC
agrep	3465
animate	2535
archie	2918
ce	4238
combine	1067
convert	1367
display	4898
fill	324
forwarder	189
fplan	2192
ghostview	8475
gzip	5220
import	1131
indent	4660
lander	825
man	626
mogrify	1374
montage	1805

PROGRAM	LOC
othello	860
par	1430
rs	343
sail	4147
sc	8343
segment	1749
stat	155
utree	9413
uuconvert	121
xalarm	2028
xasteroids	792
xcalender	1722
xdl	371
xgetftp	2653
xgopher	11769
xnlock	708
xpostit	973
xtalk	230

APPENDIX D

OUTPUT OF FUNCTION POINT METRIC FOR
ALL PROGRAMS IN THE TEST SUITE
OBTAINED BY THE COSMOS SOFTWARE TOOL [Henry 97]

UNADJUSTED FUNCTION POINT OBTAINED BY THE COSMOS TOOL

PROGRAM	EI	EO	ILF	EIF	EQ	UFP
agrep	10	8	24	0	0	42
animate	7	7	17	0	0	31
archie	7	9	17	5	0	38
ce	10	9	24	0	7	50
combine	3	4	7	0	0	14
convert	4	5	7	0	0	16
display	13	13	25	5	3	59
fill	3	4	0	0	0	7
forwarder	3	4	0	0	0	7
fplan	6	4	17	0	0	27
ghostview	16	16	32	22	16	102
gzip	16	16	17	5	10	64
import	3	4	7	5	0	19
indent	20	20	17	0	0	57
lander	3	4	7	0	0	14
man	3	4	0	5	0	12
mogrify	7	4	7	0	0	18
montage	10	5	7	0	0	22

UNADJUSTED FUNCTION POINT OBTAINED BY THE COSMOS TOOL
(cont.)

PROGRAM	EI	EO	ILF	EIF	EQ	UFP
othello	3	4	7	0	0	14
par	7	4	7	0	0	18
rs	3	4	0	0	0	7
sail	10	20	17	0	3	50
sc	23	25	39	0	16	103
segment	7	9	7	0	0	23
stat	0	0	0	5	0	5
utree	23	25	39	12	16	115
uuconvert	3	0	0	0	0	3
xalarm	7	4	7	5	3	26
xasteroids	3	4	0	0	3	10
xcalender	3	4	7	5	3	22
xdl	3	4	0	0	0	7
xgetftp	3	4	7	22	0	36
xgopher	16	16	57	17	16	122
xnlock	3	4	7	0	0	14
xpostit	3	4	7	0	0	14
xtalk	3	4	0	0	0	7

GENERAL SYSTEM CHARACTERISTICS OBTAINED BY THE COSMOS TOOL

PROGRAM	THE FOURTEEN GENERAL SYSTEM CHARACTERISTICS														GSC
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
agrep	0	0	1	0	0	0	0	0	0	1	1	1	0	0	.69
animate	0	0	1	1	1	0	0	0	1	1	0	0	0	0	.70
archie	0	1	0	2	1	0	2	0	0	0	0	0	0	0	.71
ce	0	0	0	1	0	0	0	0	0	3	1	1	0	0	.71
combine	0	0	1	0	0	0	0	0	0	0	1	1	0	0	.68
convert	0	0	1	0	0	0	0	0	0	1	0	0	0	0	.67
display	0	1	1	0	1	0	1	0	1	1	1	0	0	0	.72
fill	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.66
forwarder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.65
fplan	0	0	1	1	1	0	1	0	1	1	1	1	0	0	.73
ghostview	0	2	2	0	0	0	0	0	2	1	1	1	0	0	.74
gzip	0	0	1	1	0	0	0	0	0	3	1	1	0	1	.73
import	0	0	1	0	0	0	0	0	0	1	0	0	0	0	.67
indent	0	0	3	0	0	0	0	0	0	3	0	0	0	0	.71
lander	0	0	1	0	0	0	0	0	0	2	0	0	0	0	.68
man	0	0	1	0	0	0	0	0	0	2	0	0	0	0	.68
mogrify	0	0	2	2	0	0	0	0	1	2	0	0	0	0	.72
montage	0	0	2	2	0	0	0	0	1	2	0	0	0	0	.72

GENERAL SYSTEM CHARACTERISTICS OBTAINED BY THE COSMOS TOOL
(cont.)

PROGRAM	THE FOURTEEN GENERAL SYSTEM CHARACTERISTICS														GSC
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
othello	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.66
par	0	0	2	1	0	0	0	0	0	1	1	1	0	0	.71
rs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.65
sail	0	0	1	0	0	0	1	0	0	0	1	1	0	0	.69
sc	0	0	2	1	0	0	1	0	0	2	1	1	0	1	.74
segment	0	0	2	2	0	0	0	0	0	0	0	0	0	0	.69
stat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.65
utree	0	0	1	0	0	0	0	0	0	1	1	2	2	0	.72
uuconvert	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.65
xalarm	0	1	1	0	0	0	0	0	0	2	1	1	0	0	.71
xasteroids	0	0	1	0	0	0	1	0	0	0	0	0	0	0	.67
xcalender	0	1	2	0	0	0	0	1	1	2	1	1	0	0	.74
xdl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.65
xgetftp	1	2	2	1	3	2	1	0	0	0	0	0	0	0	.77
xgopher	2	2	3	1	4	0	2	0	0	2	1	0	0	0	.82
xnlock	0	0	3	0	0	0	0	0	0	0	0	0	0	0	.68
xpostit	0	0	1	0	0	0	0	0	0	0	1	1	0	0	.68
xtalk	1	1	0	0	0	1	0	0	0	0	0	0	0	0	.68

FUNCTION POINTS OBTAINED BY THE COSMOS TOOL

PROGRAM	UFP	GSC	FP	ESTIMATED LOC	ACTUAL LOC
agrep	42	.69	29	3709.4	3465
animate	31	.70	21.7	2777.6	2535
archie	38	.71	27	3453.4	2918
ce	50	.71	35.5	4544.0	4238
combine	14	.68	9.5	1218.6	1067
convert	16	.67	10.7	1372.2	1367
display	59	.72	42.5	5437.4	4898
fill	7	.66	4.6	591.4	324
forwarder	7	.65	4.6	582.4	189
fplan	27	.73	19.7	2522.9	2192
ghostview	102	.74	75.5	9661.4	8475
gzip	64	.73	46.7	5980.2	5220
import	19	.67	12.7	1629.4	1131
indent	57	.71	40.5	5180.2	4660
lander	14	.68	9.5	1218.6	825
man	12	.68	8.2	1044.5	626
mogrify	18	.72	13.0	1658.9	1374
montage	22	.72	15.8	2027.5	1805

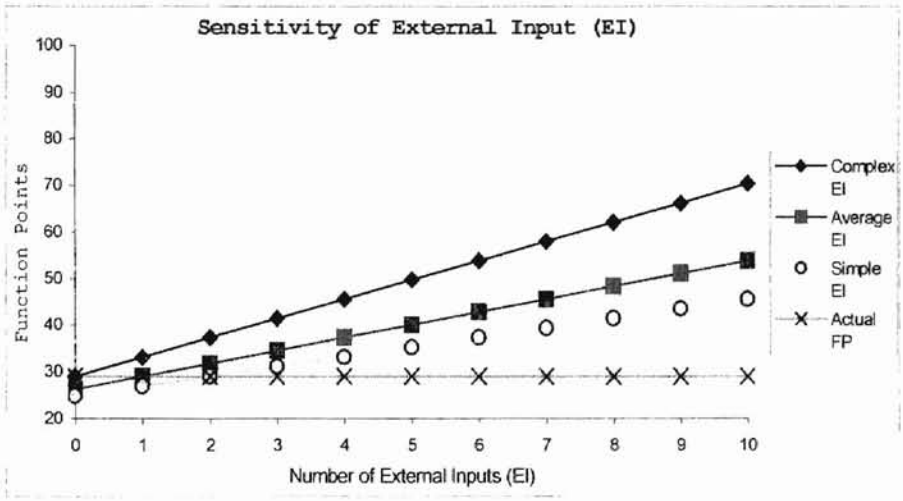
FUNCTION POINTS OBTAINED BY THE COSMOS TOOL
(cont.)

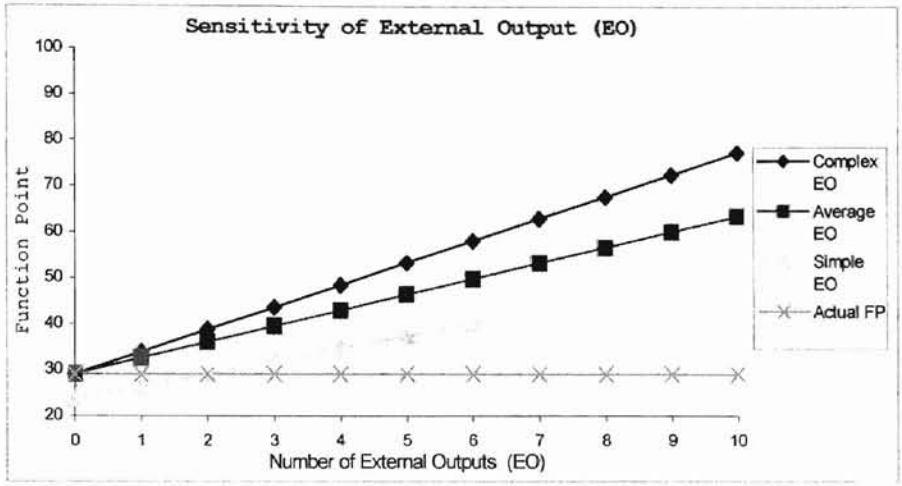
PROGRAM	UFP	GSC	FP	ESTIMATED LOC	ACTUAL LOC
othello	14	.66	9.2	1182.7	860
par	18	.71	12.8	1635.8	1430
rs	7	.65	4.6	582.4	343
sail	50	.69	34.5	4416.0	4147
sc	103	.74	76.2	9756.2	8343
segment	23	.69	15.9	2031.4	1749
stat	5	.65	3.3	416.0	155
utree	115	.72	82.8	10598.4	9413
uuconvert	3	.65	2.0	249.6	121
xalarm	26	.71	18.5	2362.9	2028
xasteroids	10	.67	6.7	857.6	792
xcalender	22	.74	16.3	2083.8	1722
xdl	7	.65	4.6	582.4	371
xgetftp	36	.77	27.7	3548.2	2653
xgopher	122	.82	100.0	12805.1	11769
xnlock	14	.68	9.5	1218.6	708
xpostit	14	.68	9.5	1218.6	973
xtalk	7	.68	4.8	609.3	230

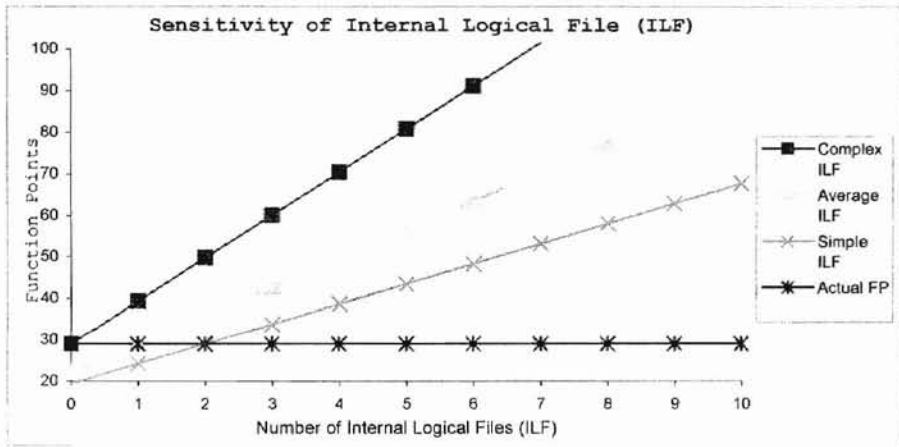
APPENDIX E

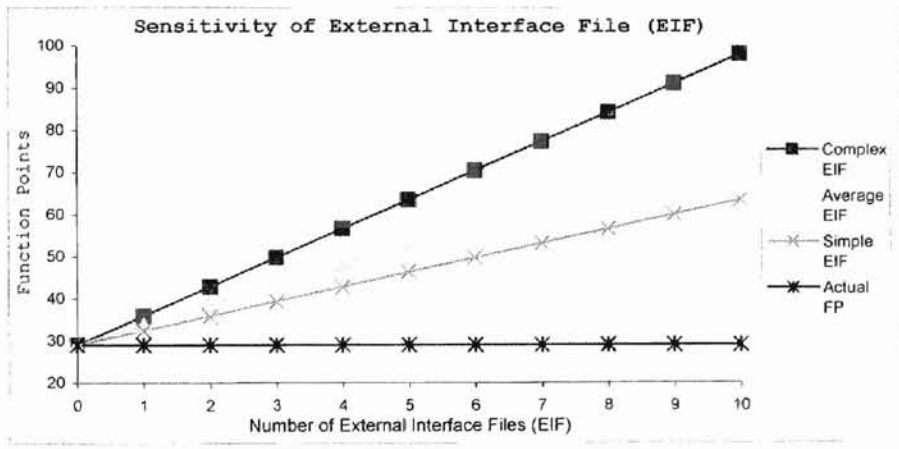
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR ALL PROGRAMS IN THE TEST SUITE

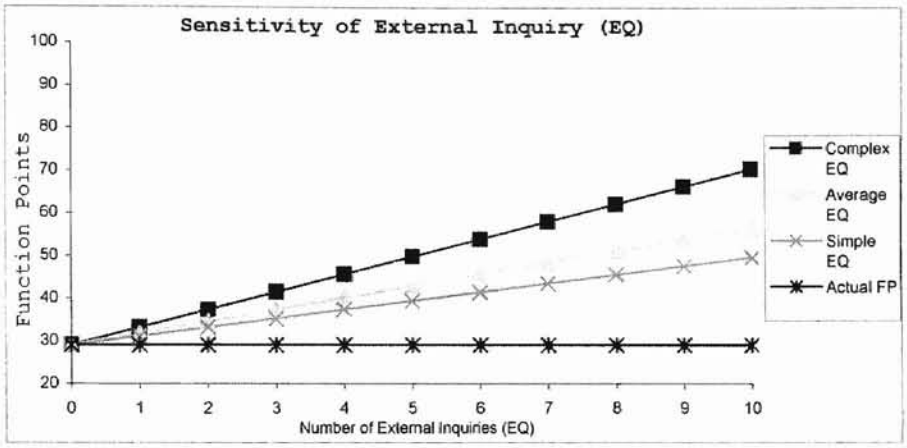
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE AGREP PROGRAM

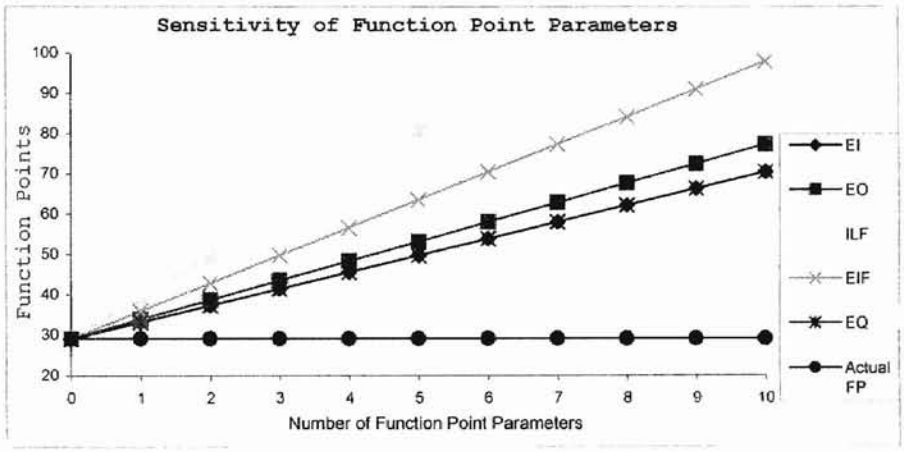


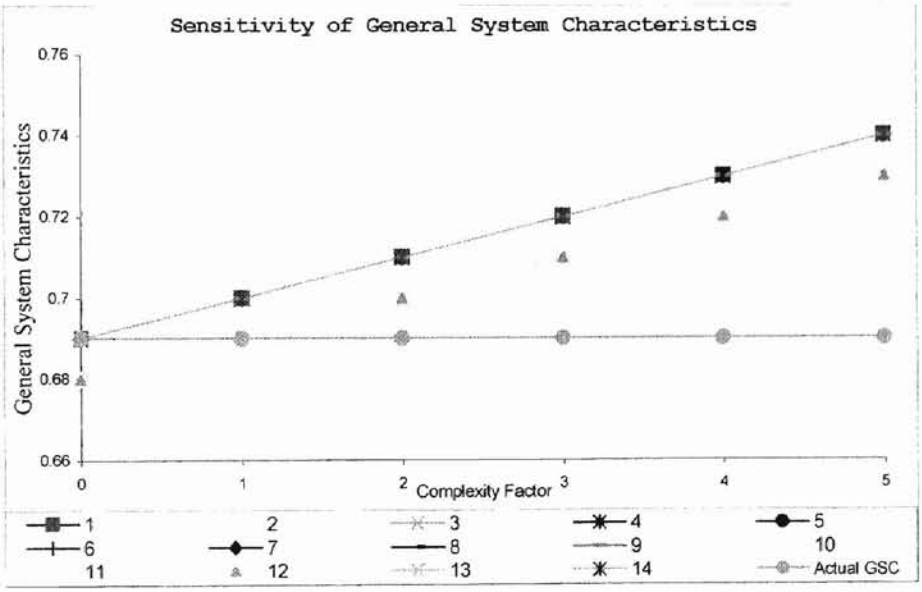

Sensitivity of External Output (EO)
Function Point
100
90
80
70
60
50
40
30
20
0 1 2 3 4 5 6 7 8 9 10
Number of External Outputs (EO)
Complex EO
Average EO
Simple EO
Actual FP


Sensitivity of Internal Logical File (ILF)
Function Points
100
90
80
70
60
50
40
30
20
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

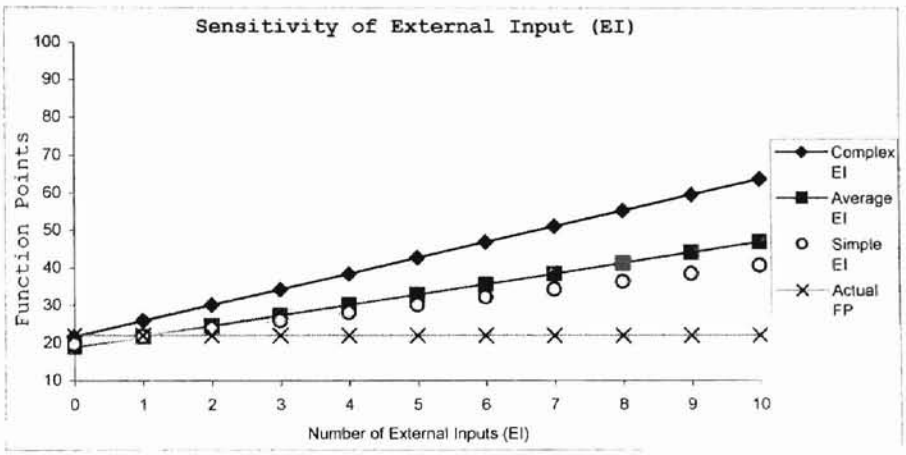

Sensitivity of External Interface File (EIF)
Function Points
100
90
80
70
60
50
40
30
20
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

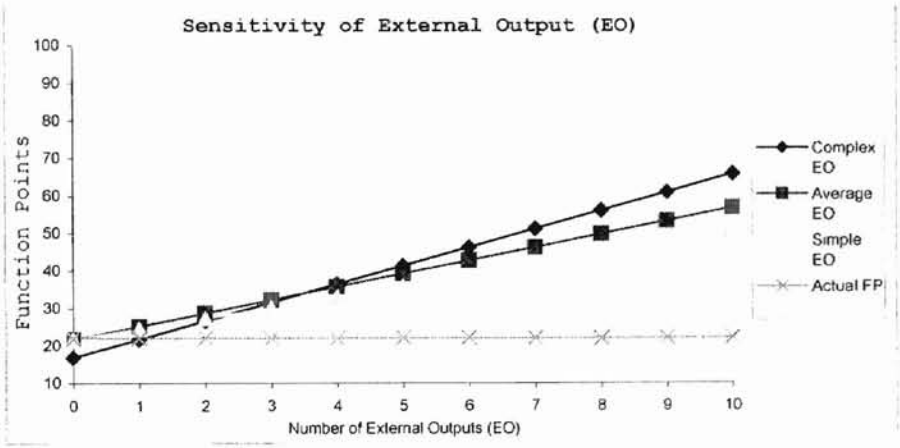

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

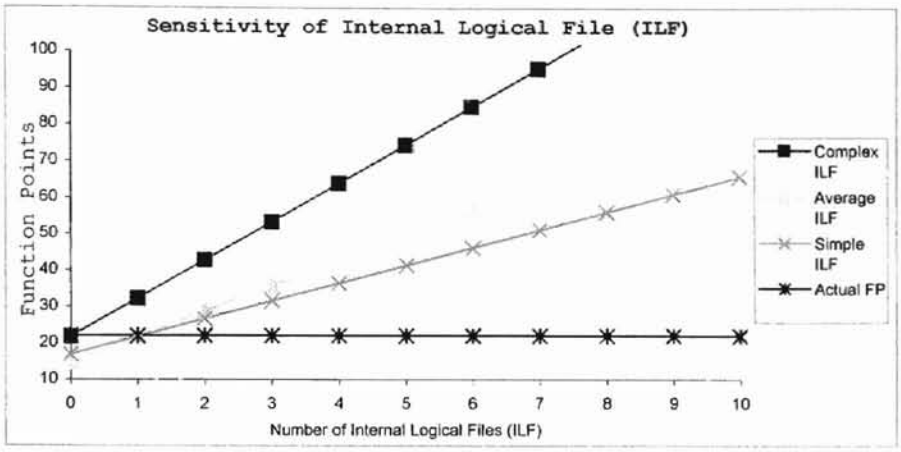

Sensitivity of Function Point Parameters
Function Points
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

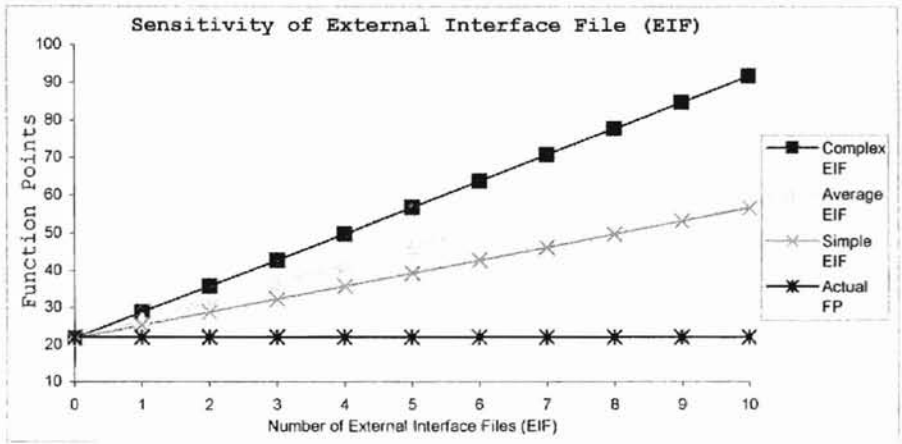

Sensitivity of General System Characteristics
General System Characteristics
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

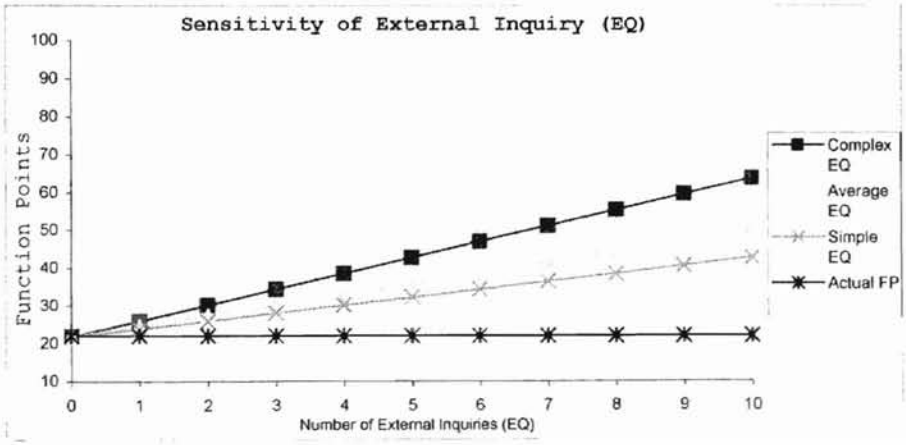
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE ANIMATE PROGRAM

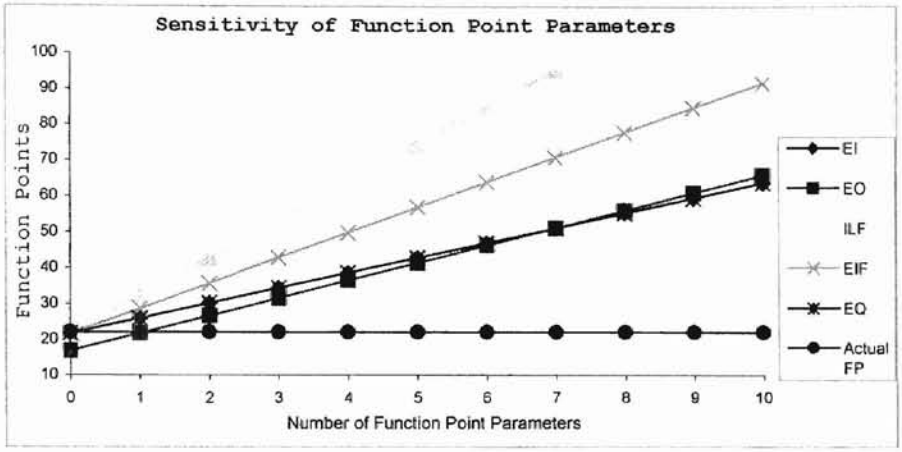


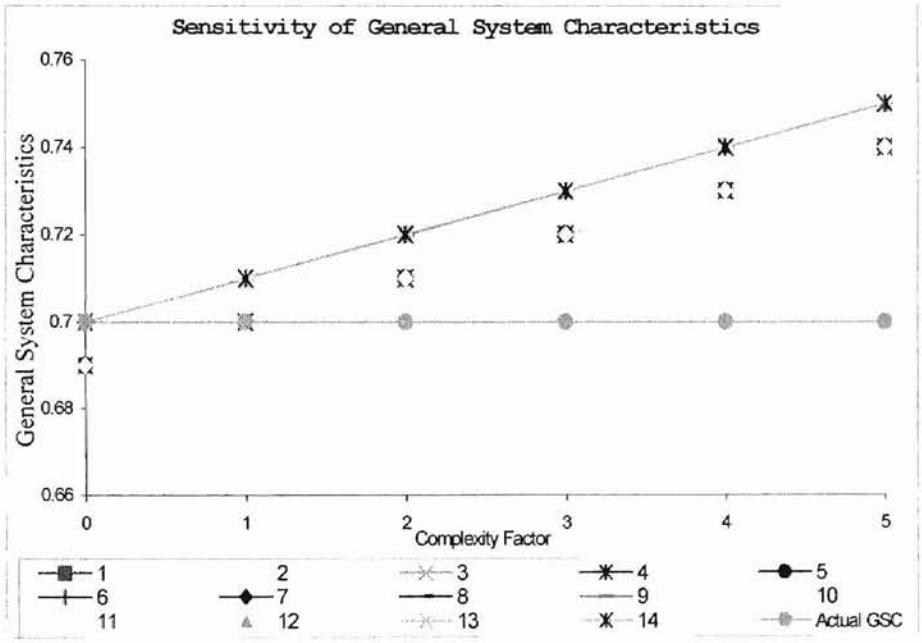



Sensitivity of Internal Logical File (ILF)
Function Points
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

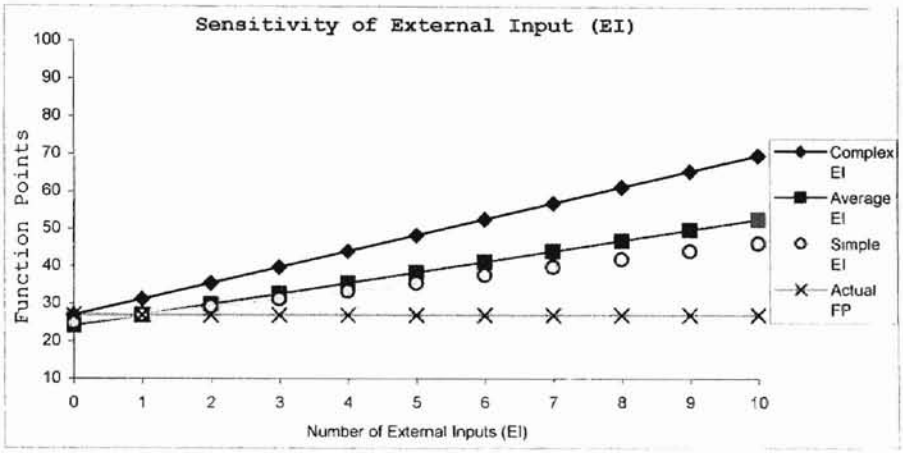

Sensitivity of External Interface File (EIF)
Function Points
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

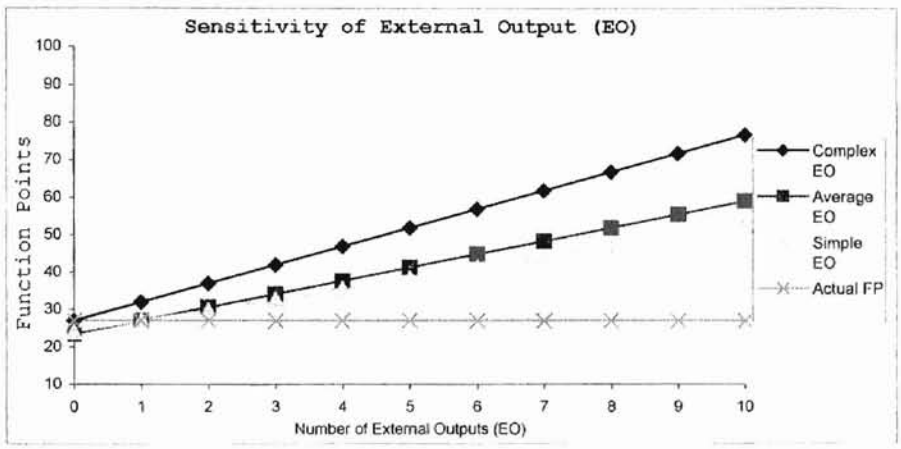

Sensitivity of External Inquiry (EQ)
Function Points
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

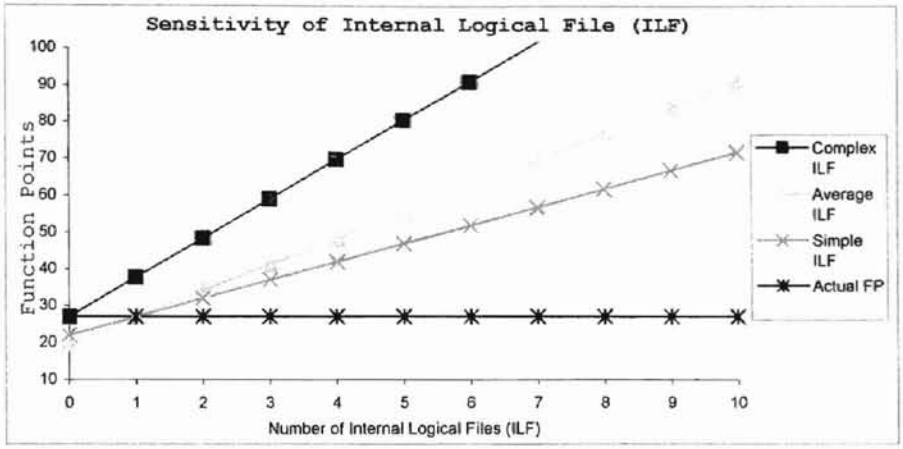

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

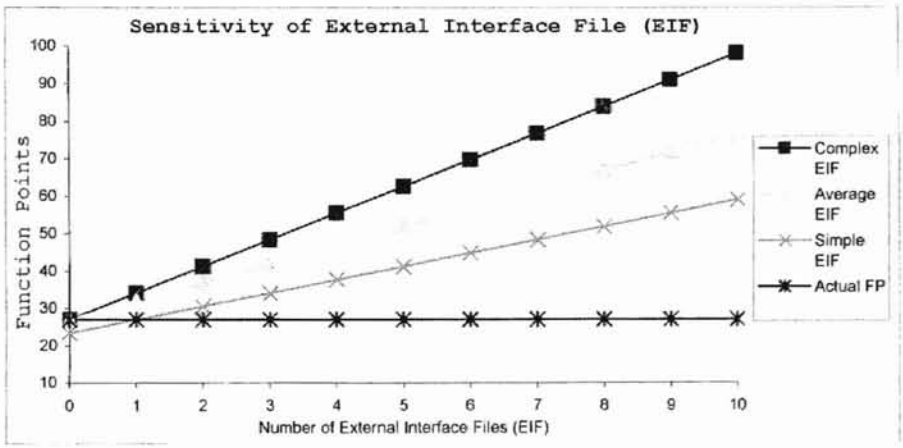

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

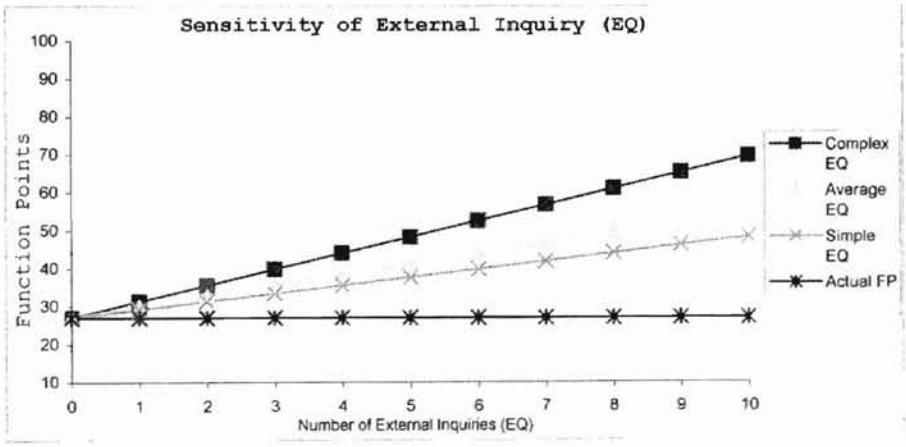
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE ARCHIE PROGRAM

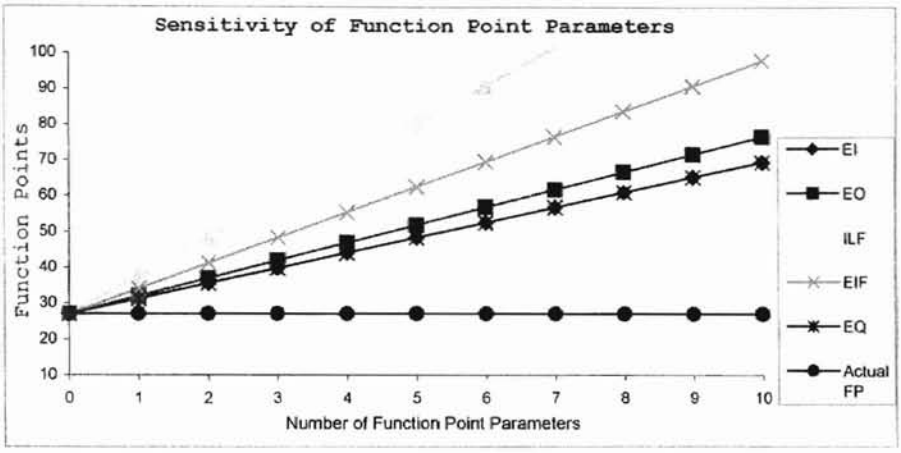


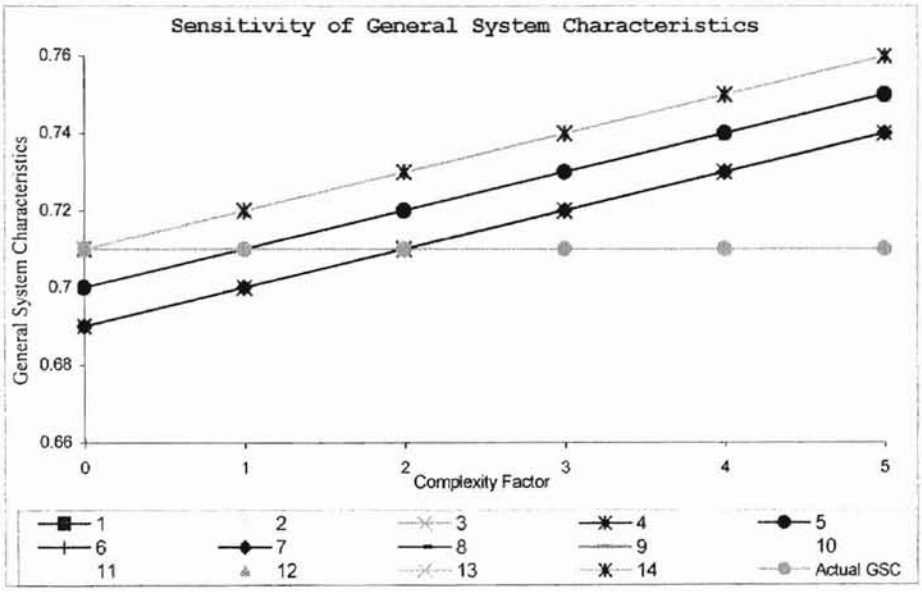



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

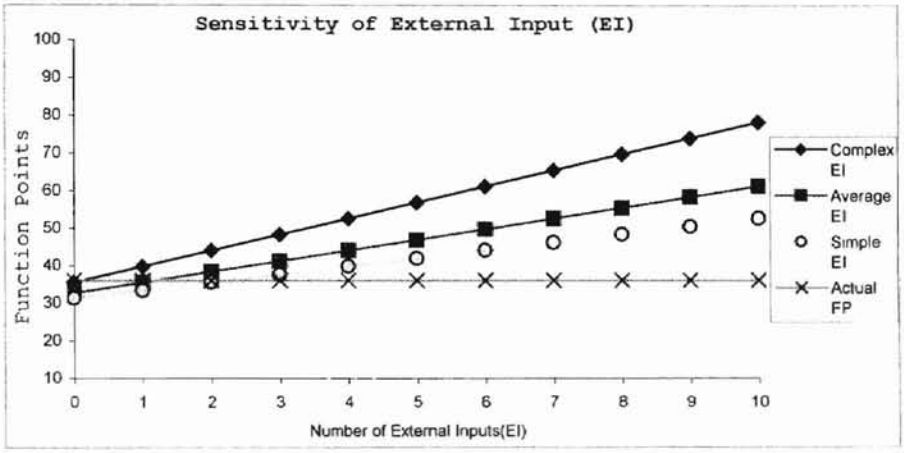

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

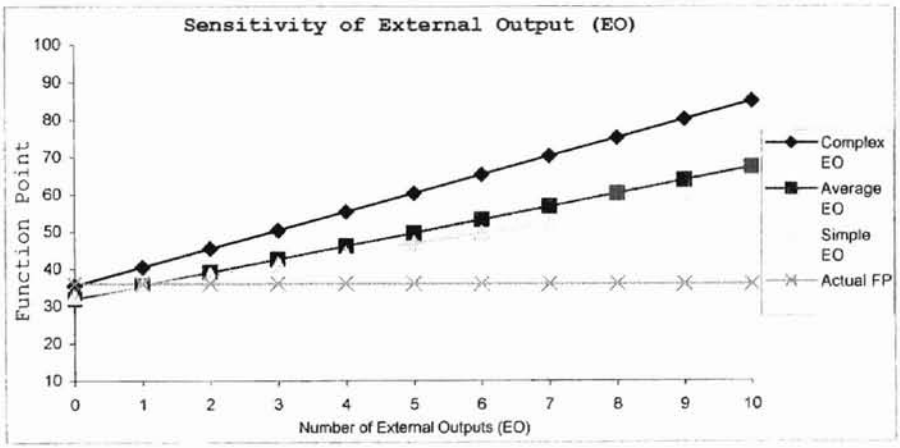

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

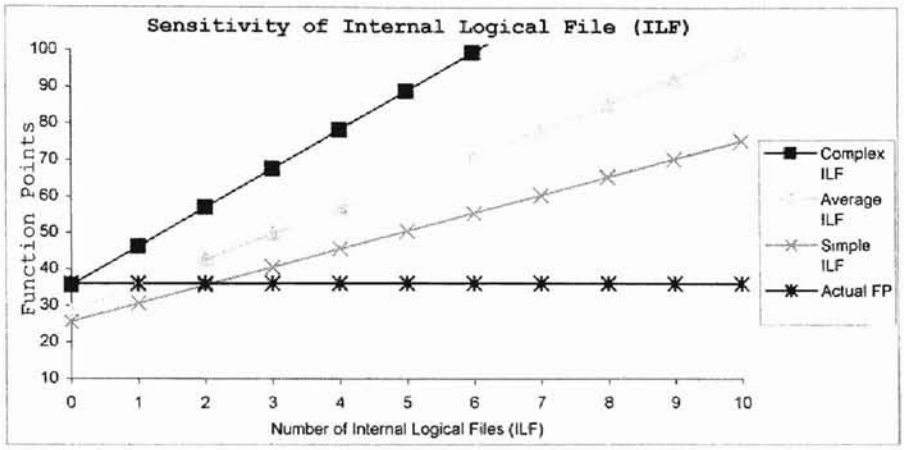

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

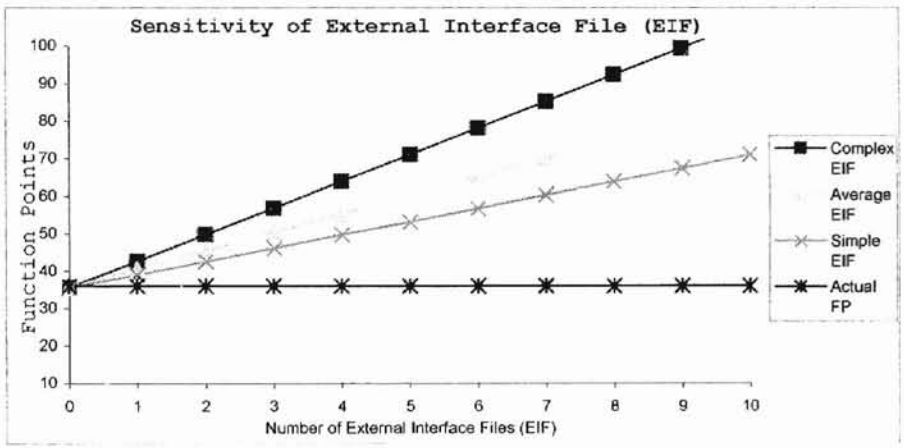

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

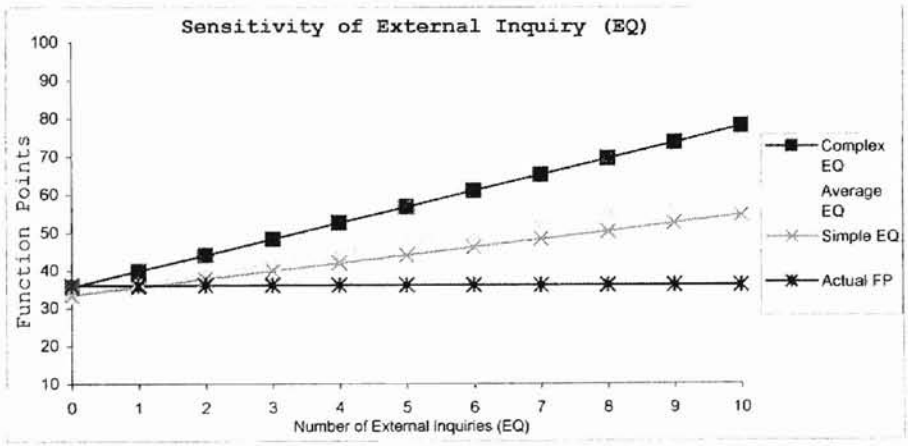
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE CE PROGRAM

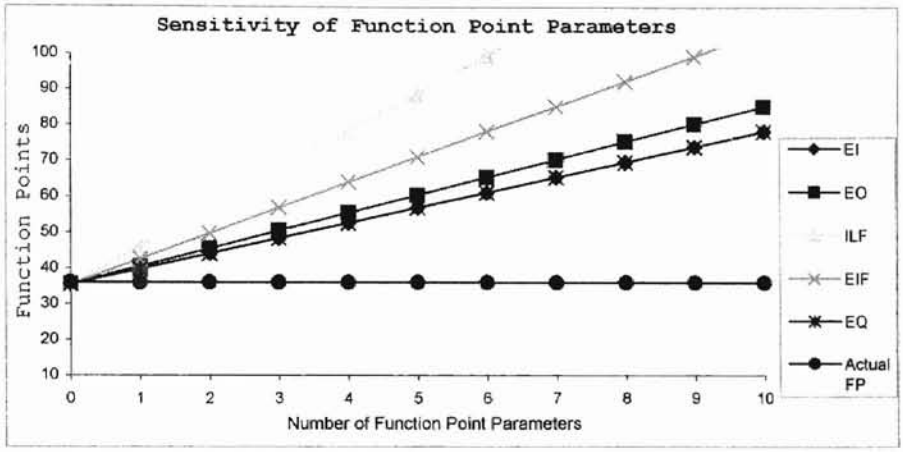


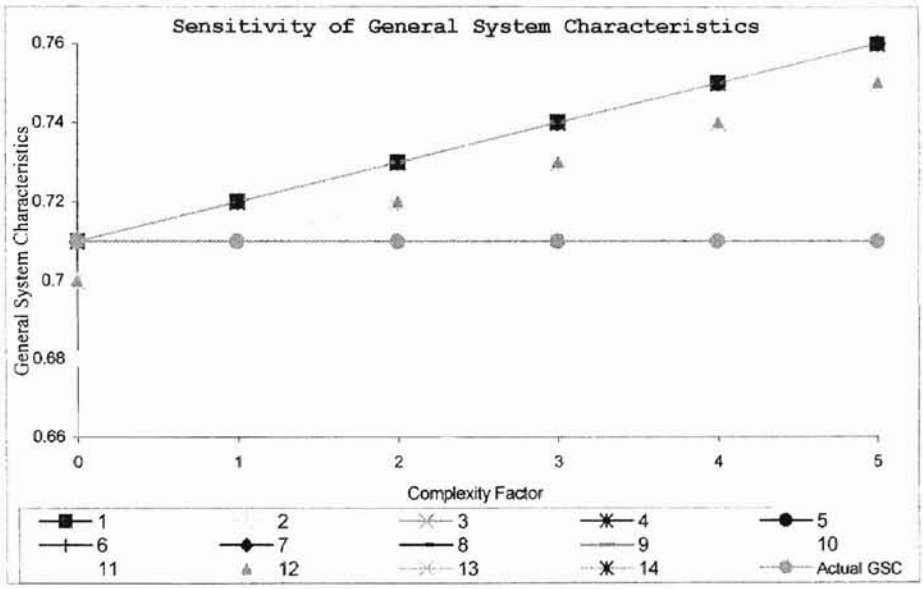



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

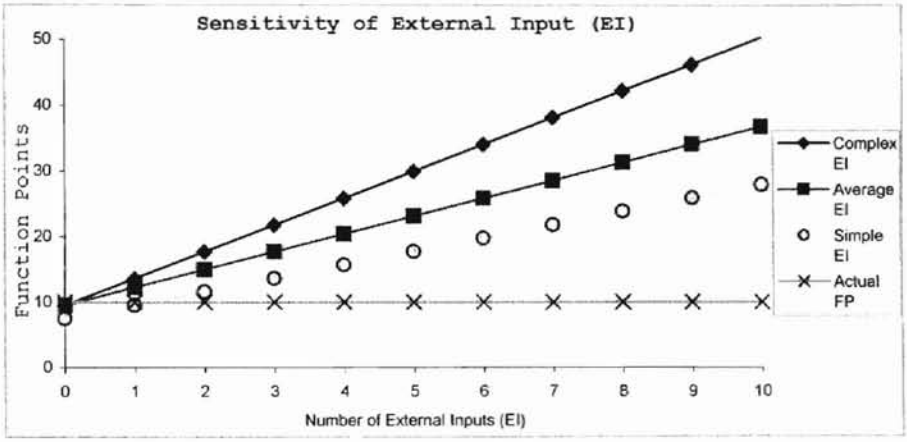

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

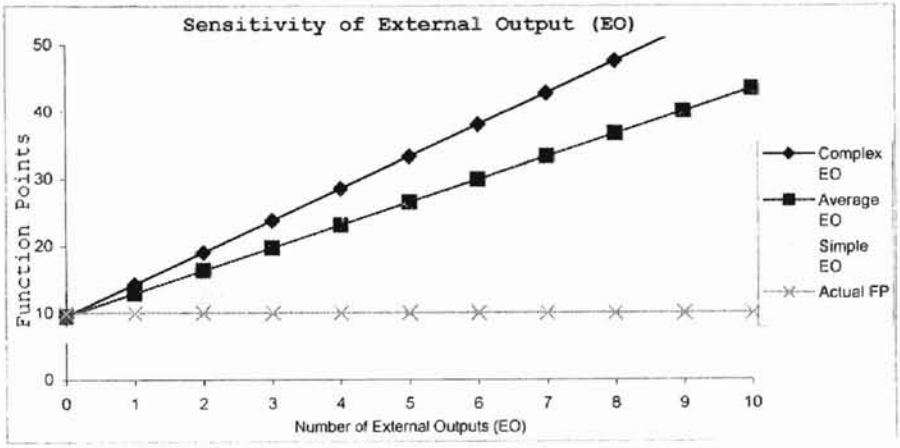

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

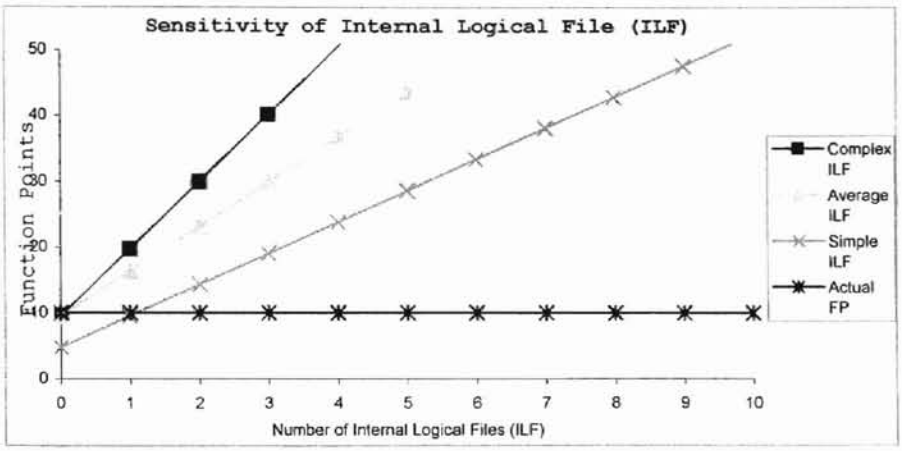

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

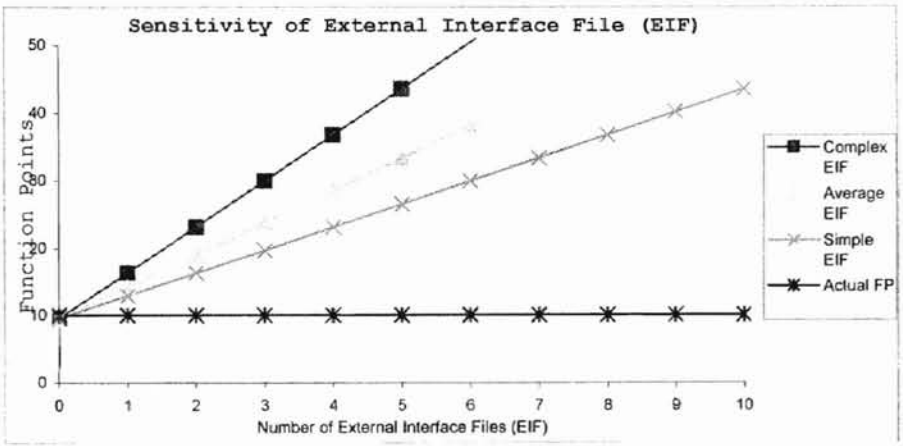

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

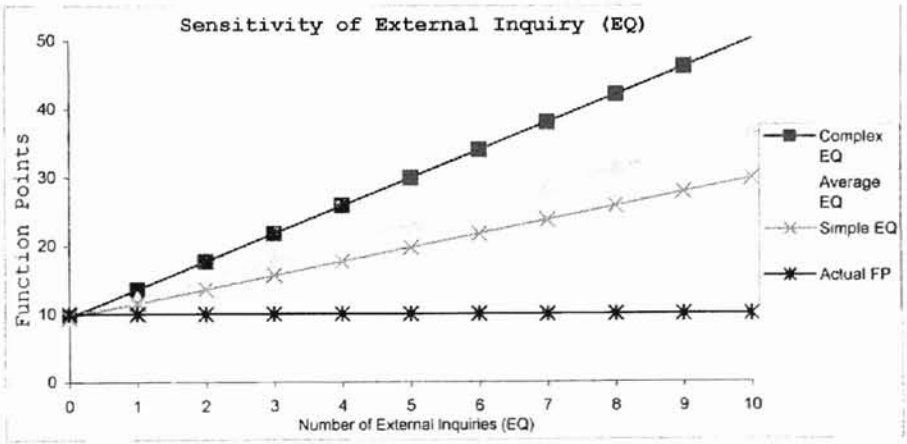
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE COMBINE PROGRAM

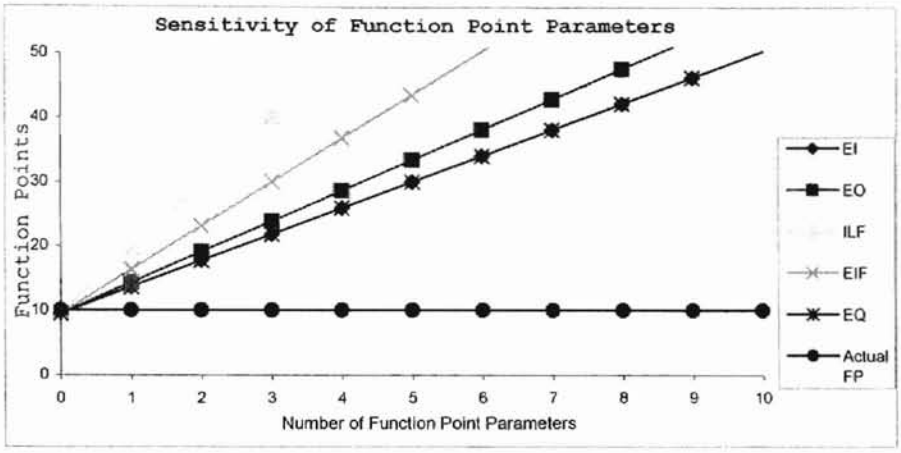


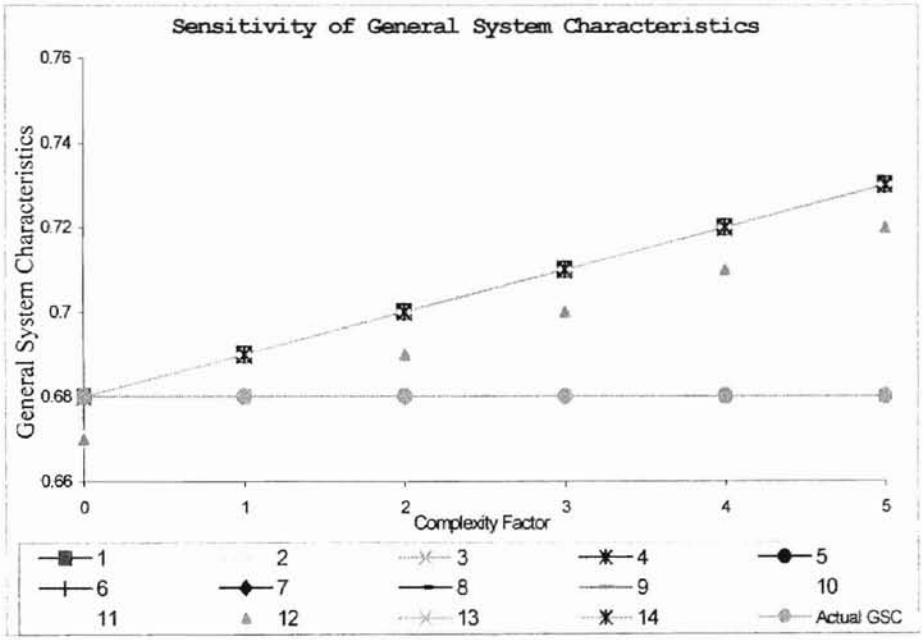



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

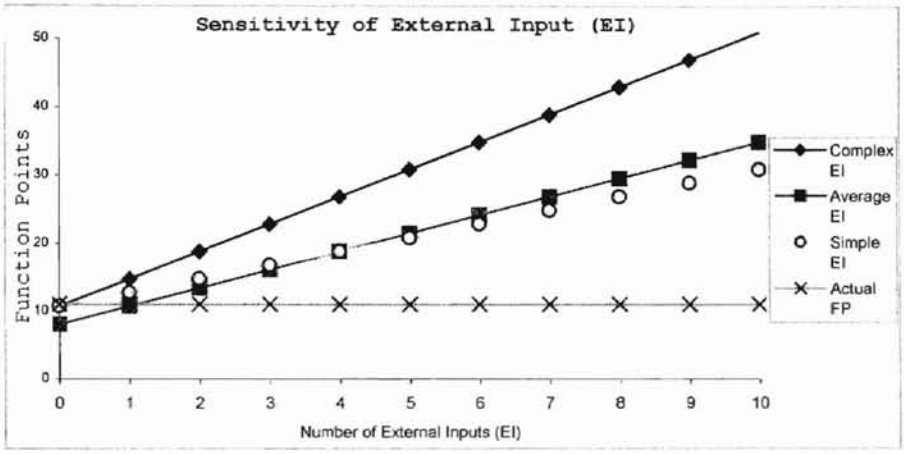

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

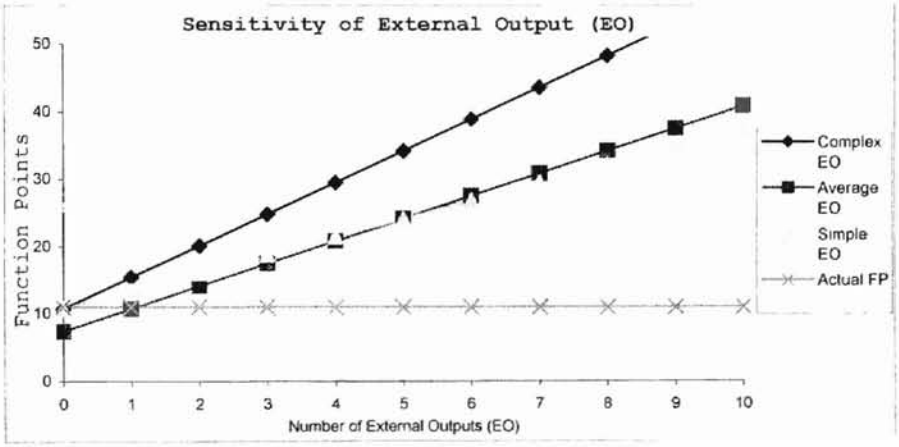

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

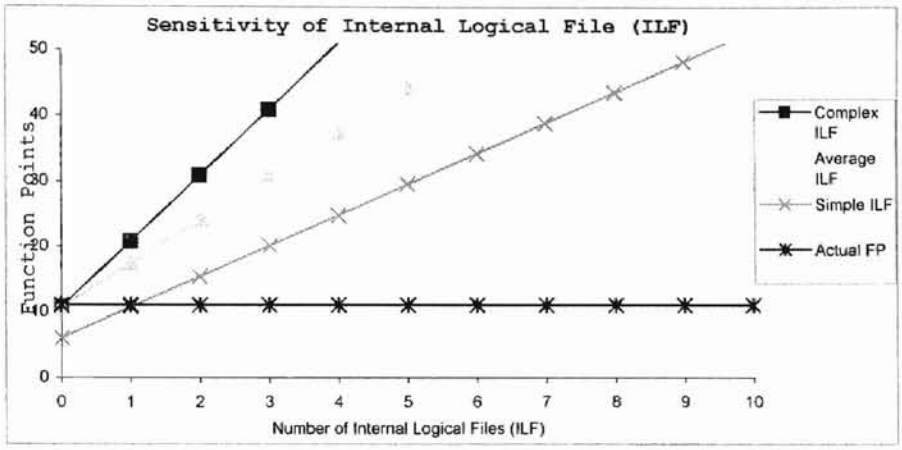

Sensitivity of Function Point Parameters
Function Points
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

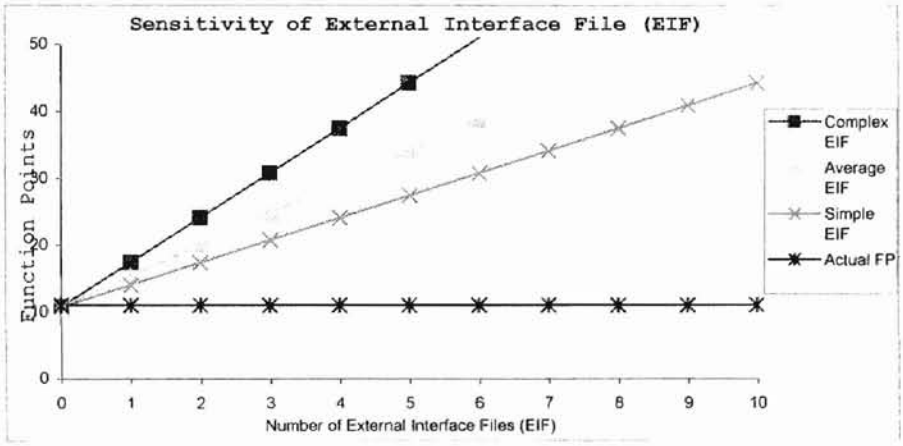

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0 1 2 3 4 5
Complexity Factor
1 2 3 4 5
6 7 8 9 10
11 12 13 14 Actual GSC

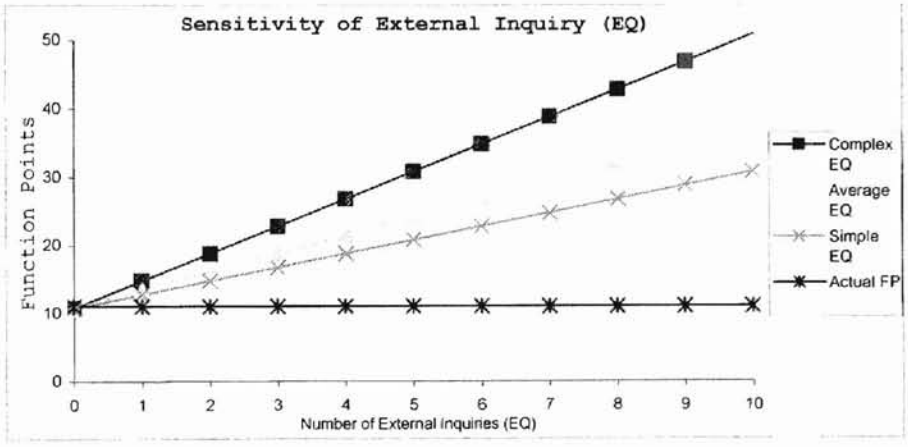
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE CONVERT PROGRAM

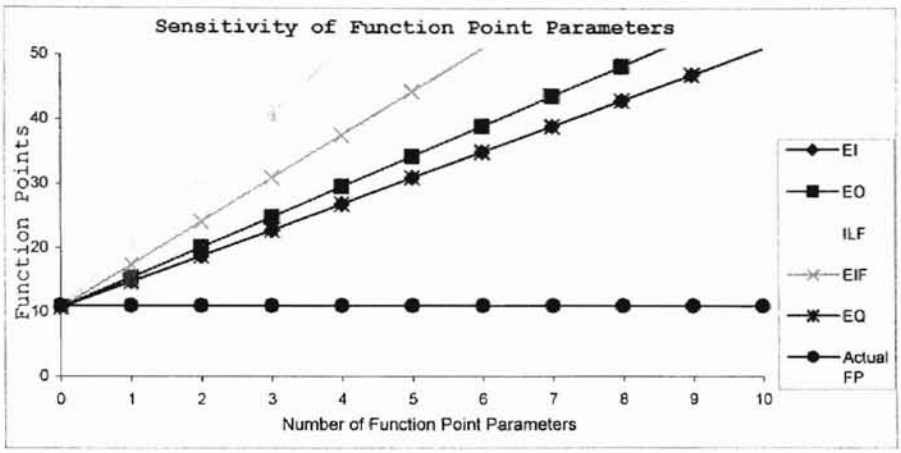


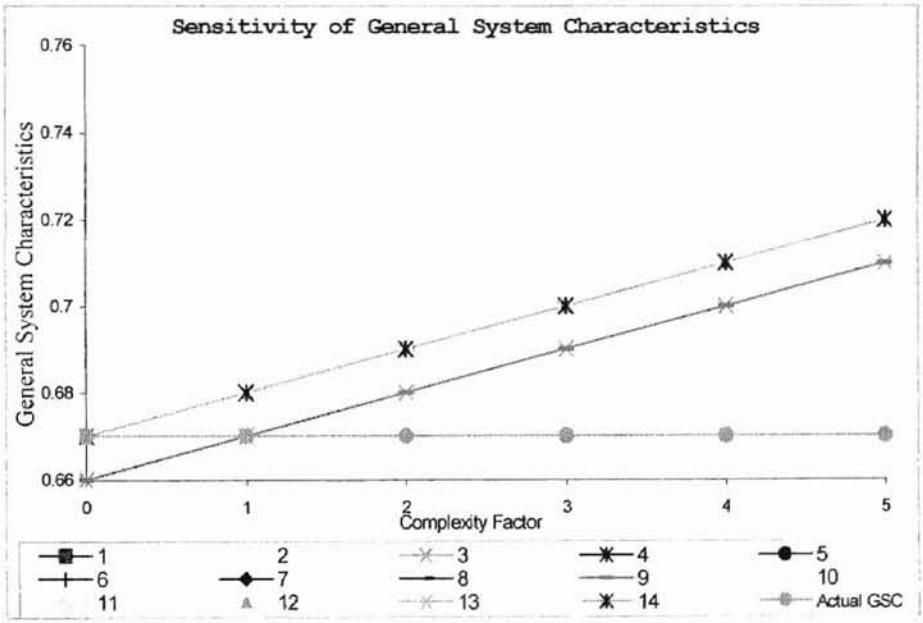



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

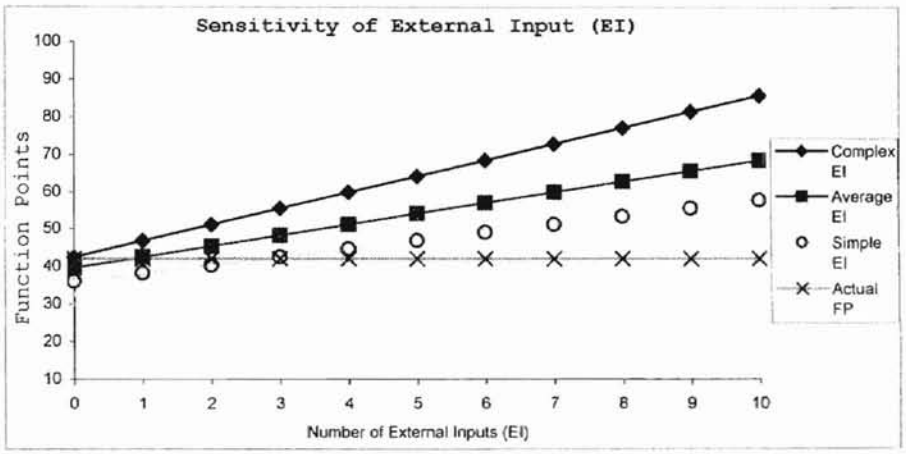

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

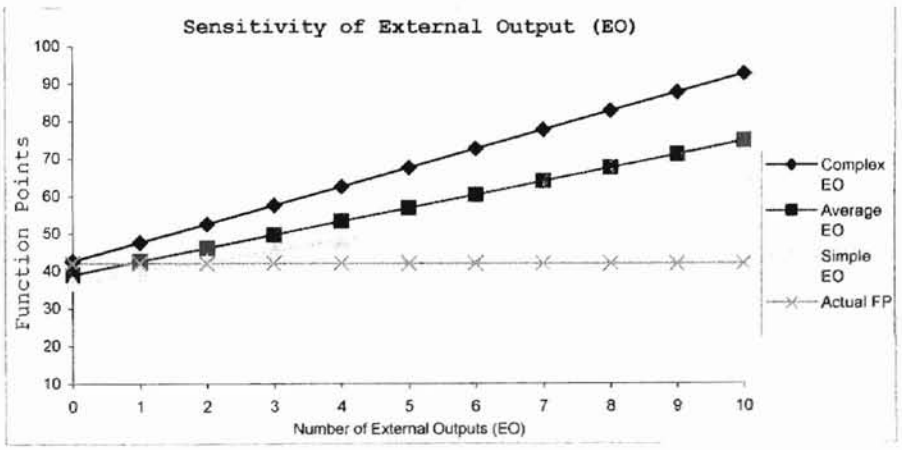

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquires (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

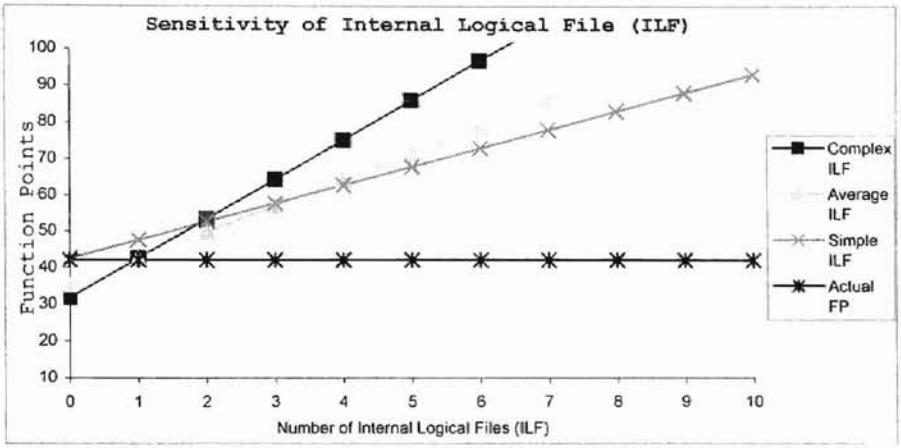

Sensitivity of Function Point Parameters
Function Points
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

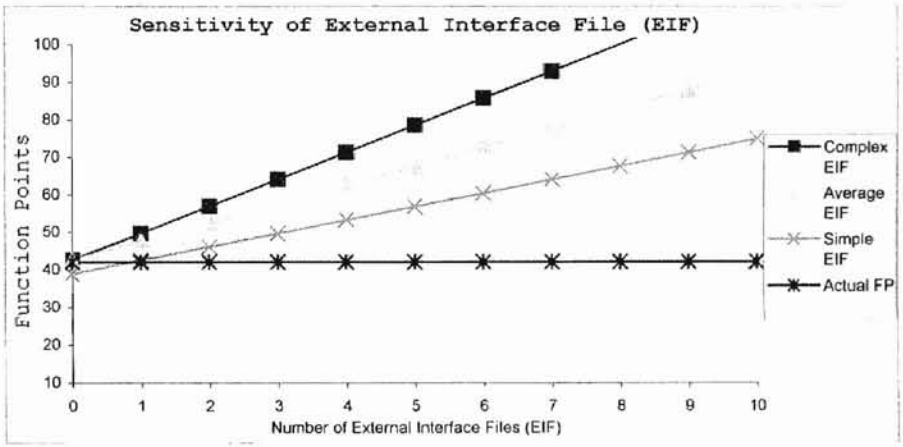

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

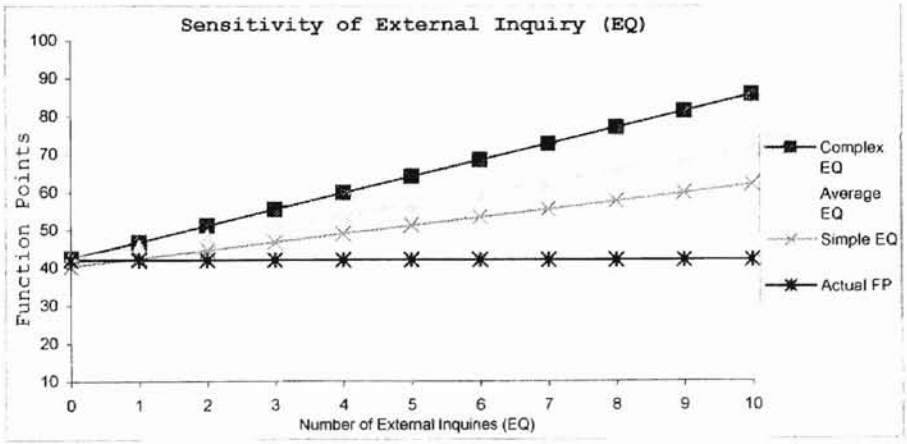
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE DISPLAY PROGRAM

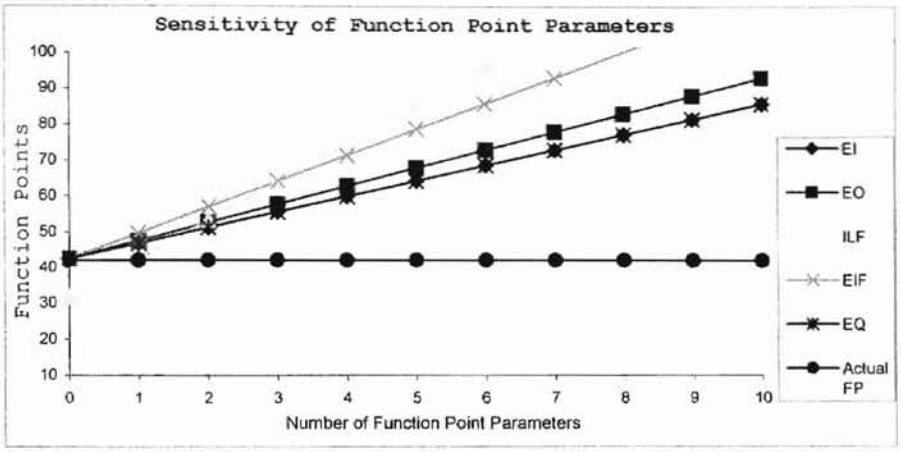


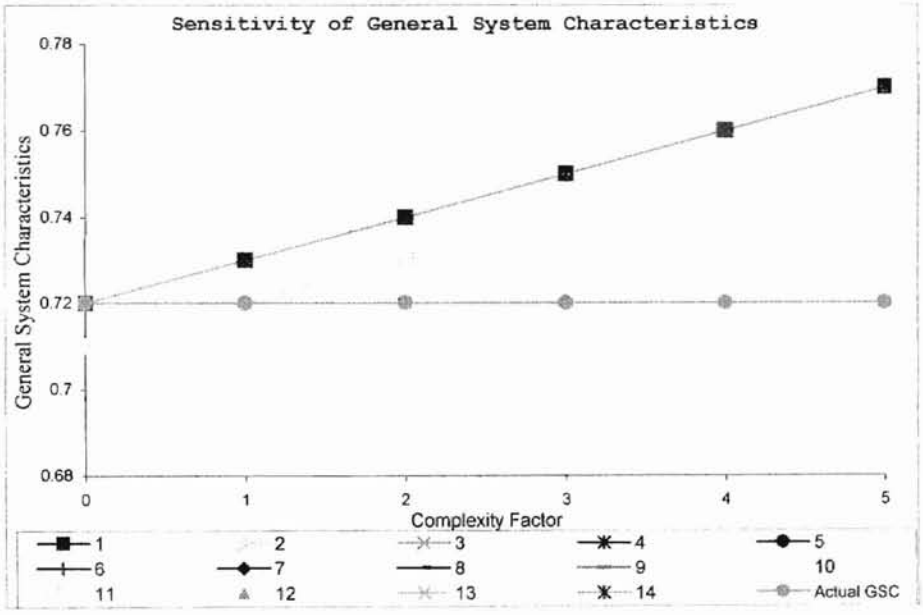



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

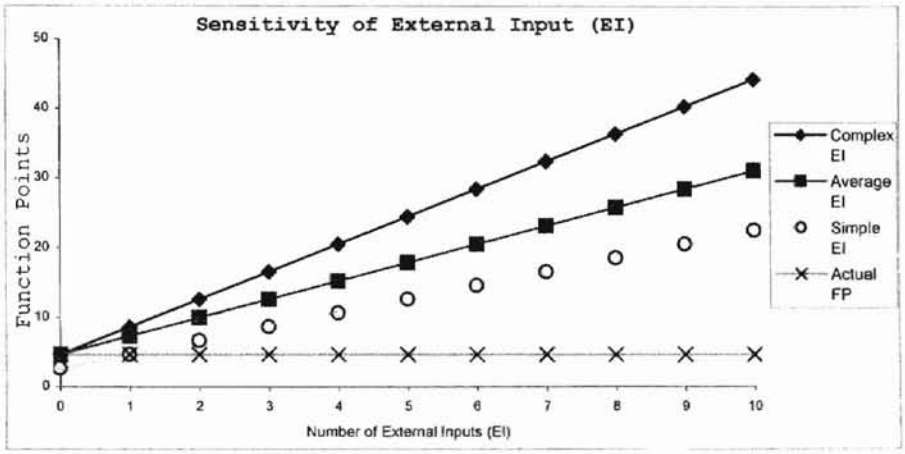

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

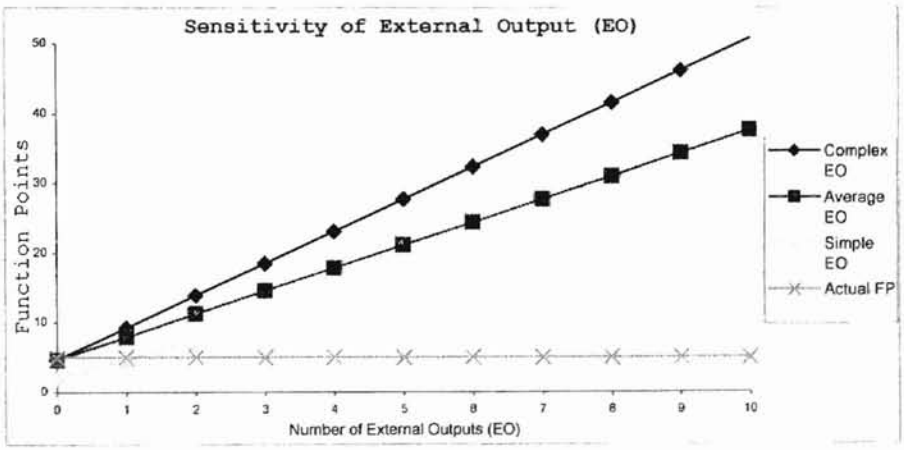

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

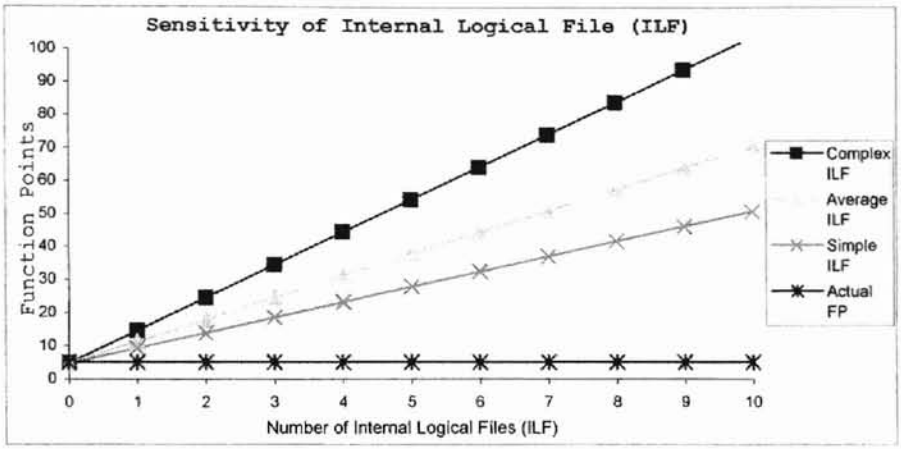

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

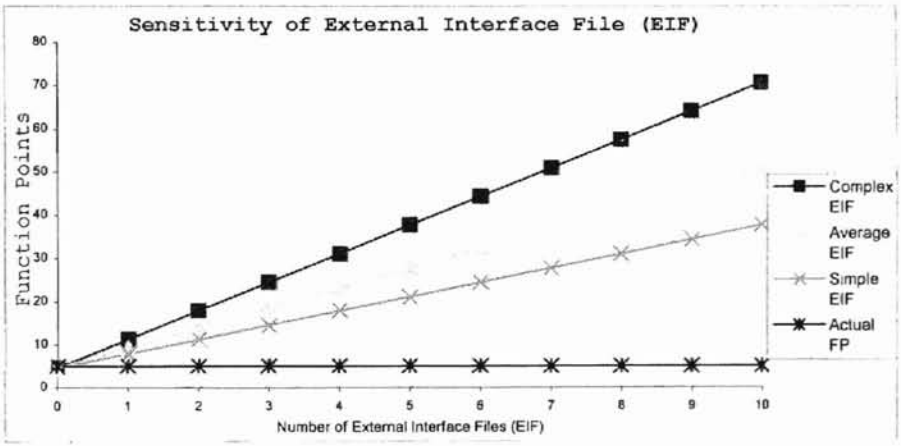

Sensitivity of General System Characteristics
General System Characteristics
0.78
0.76
0.74
0.72
0.7
0.68
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

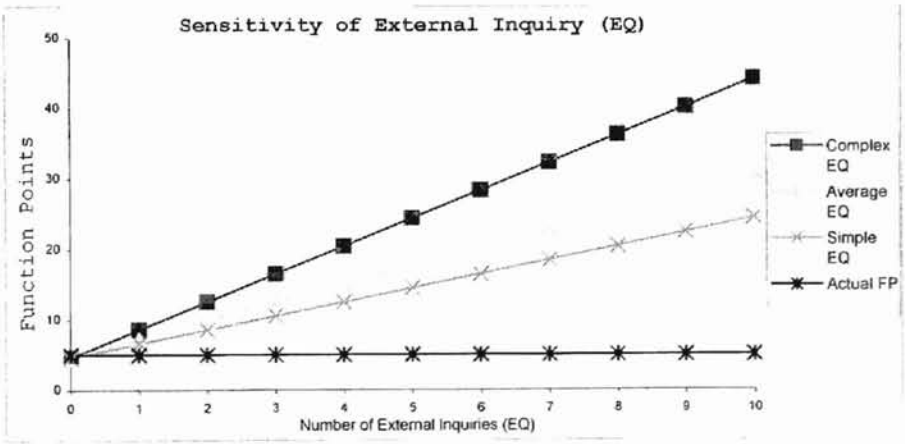
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE FILL PROGRAM

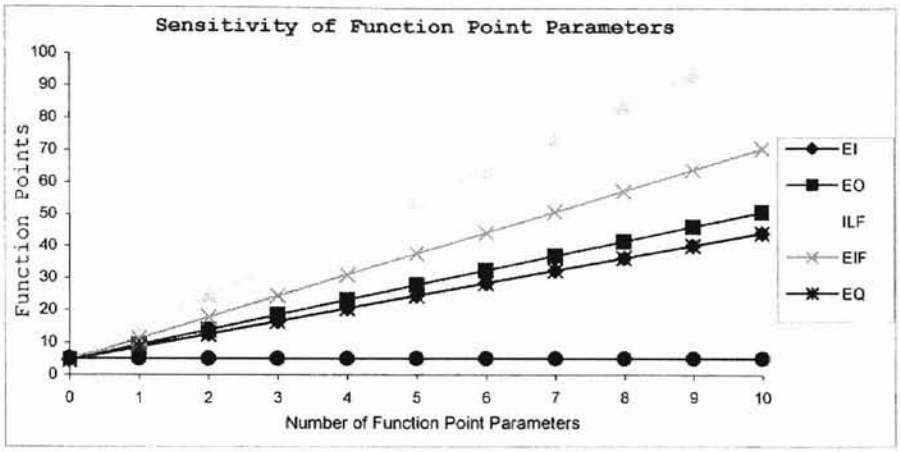


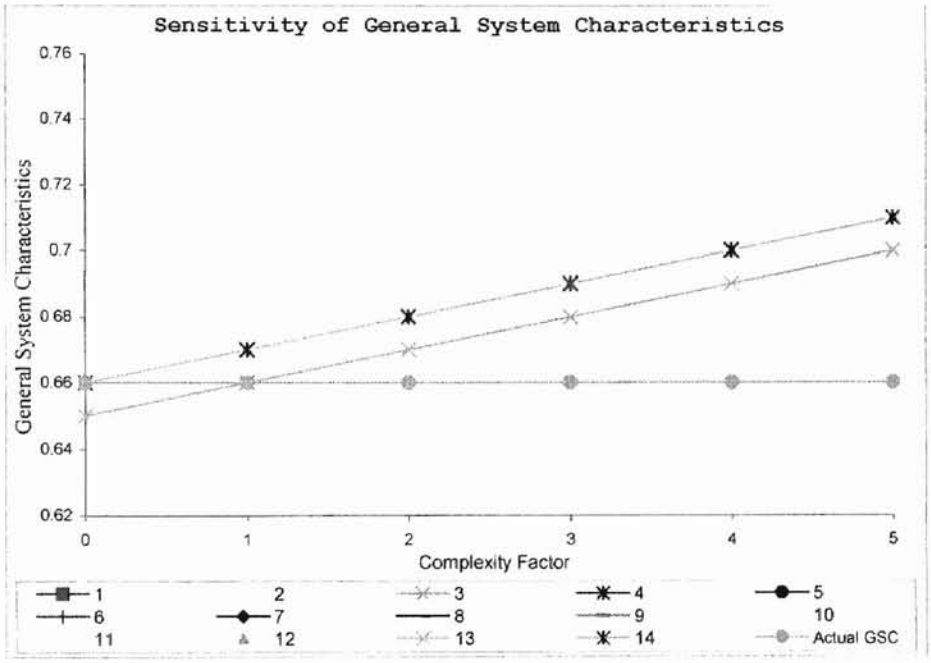



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

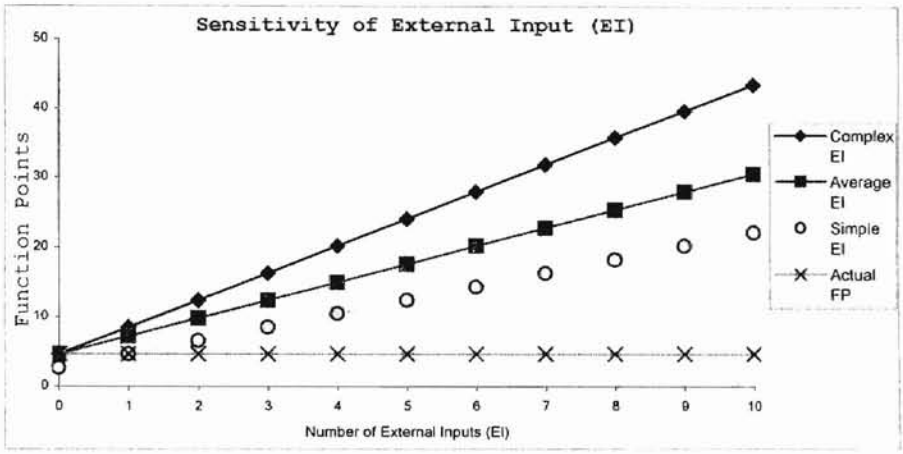

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

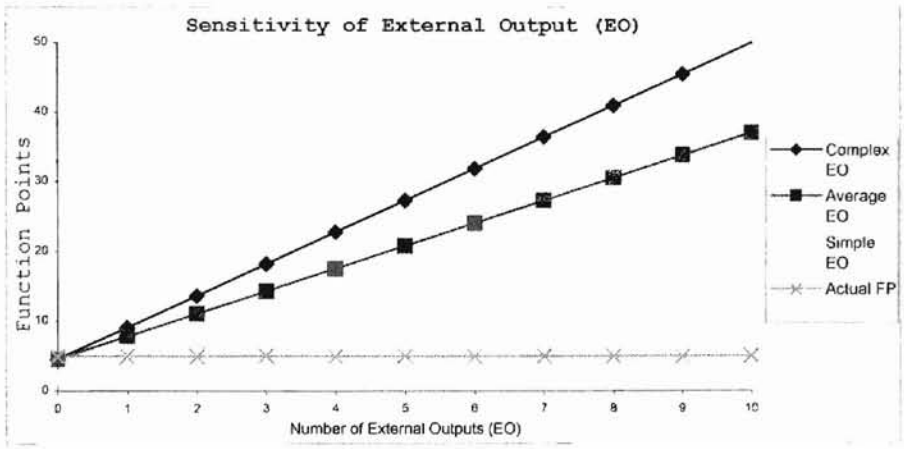

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

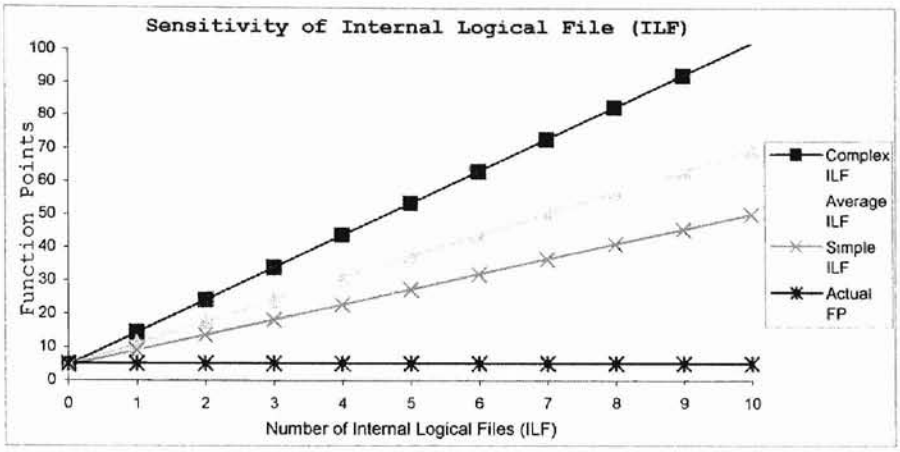

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ

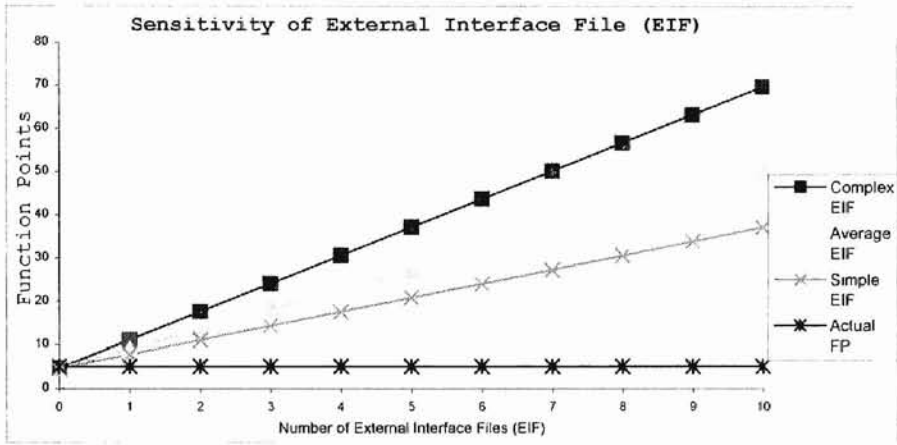

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

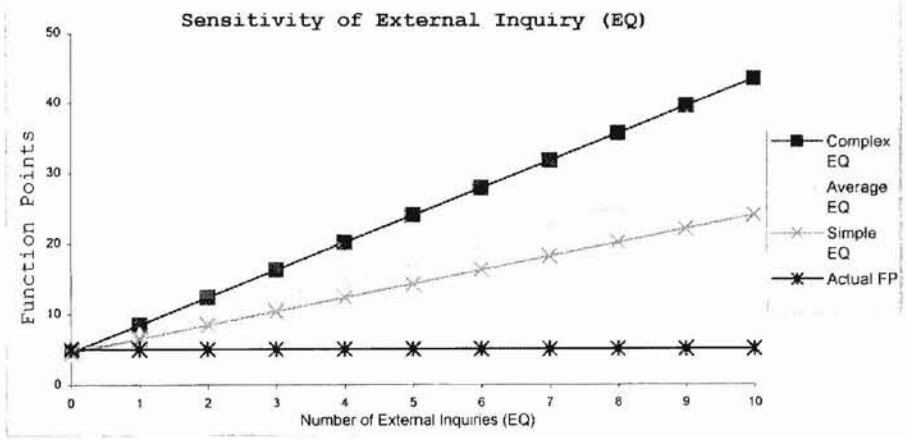
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE FORWARDER PROGRAM

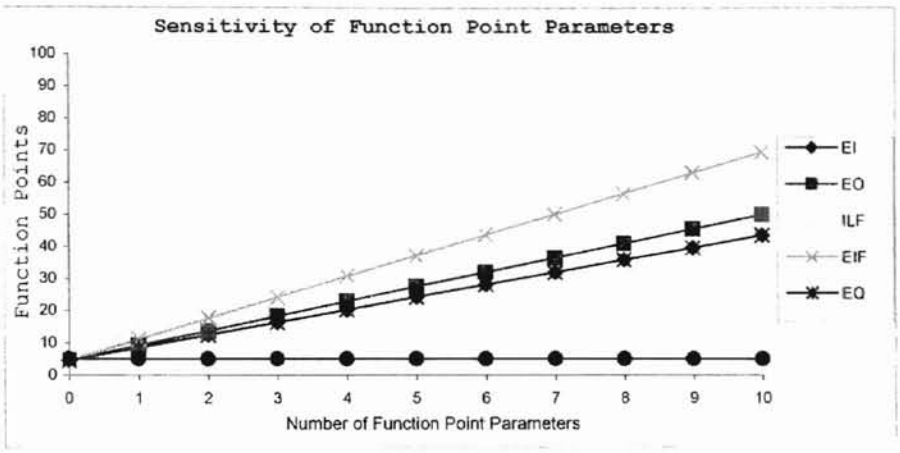


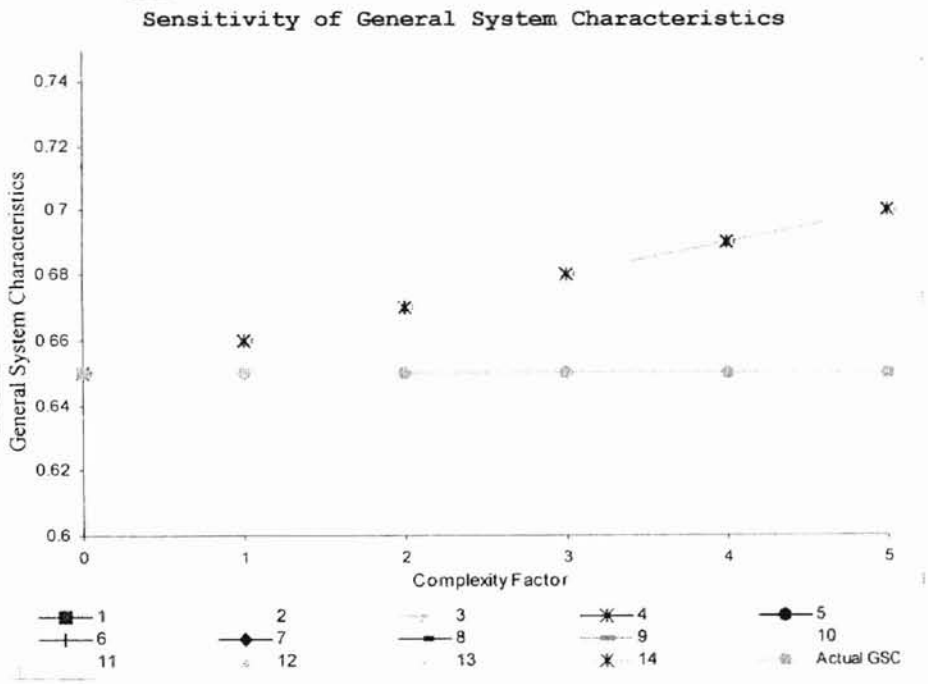



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

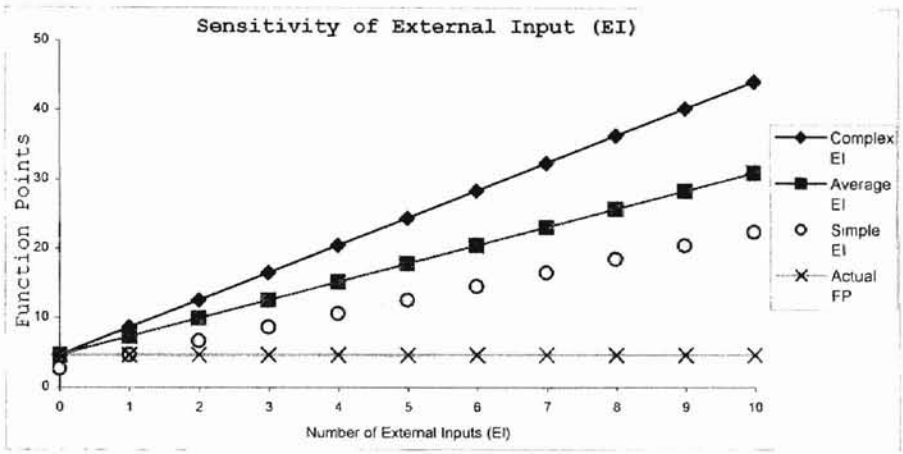

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

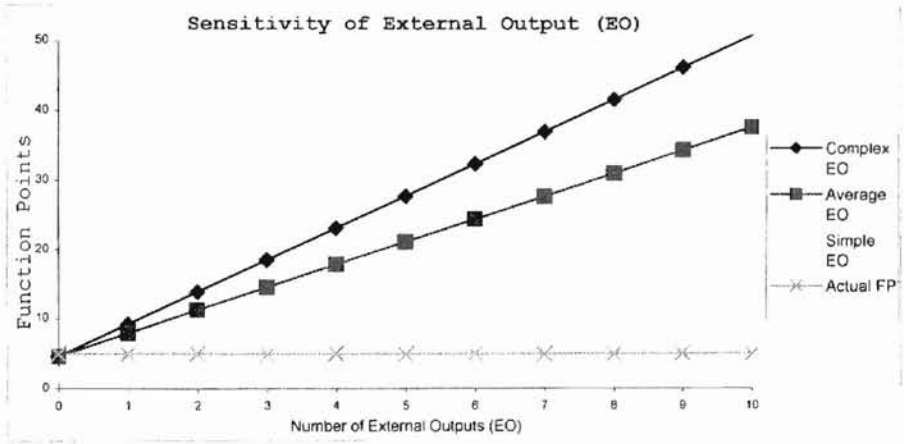

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

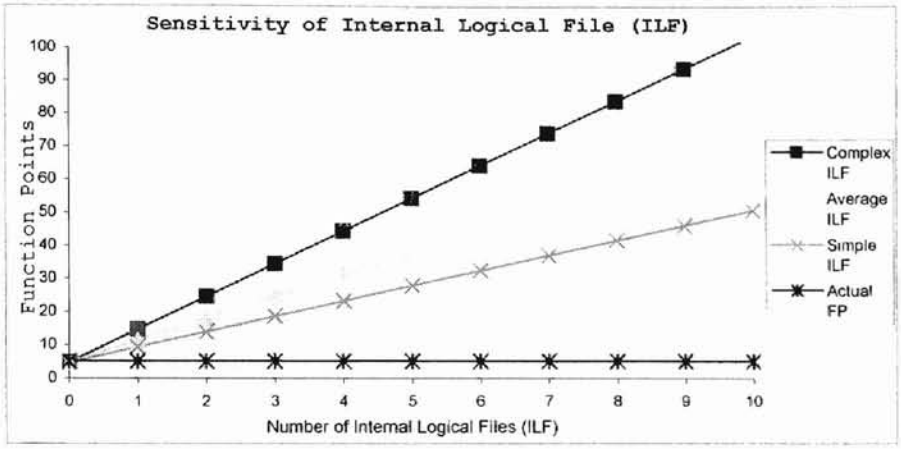

Sensitivity of Function Point Parameters
Function Points
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ

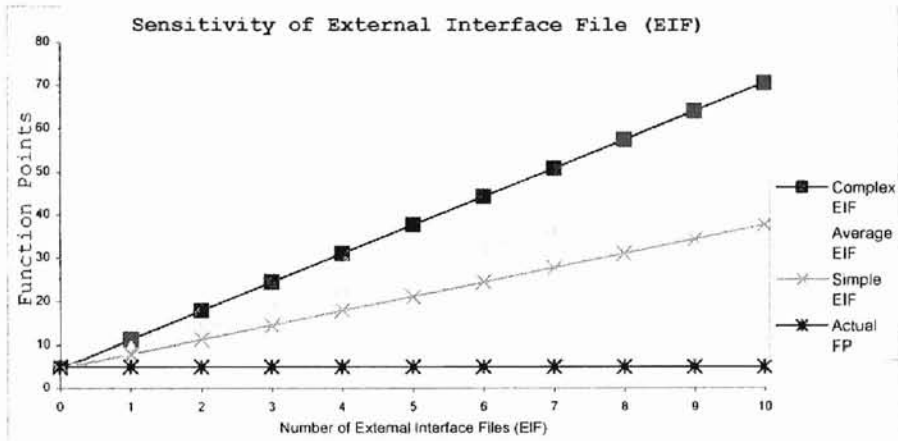

Sensitivity of General System Characteristics
General System Characteristics
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

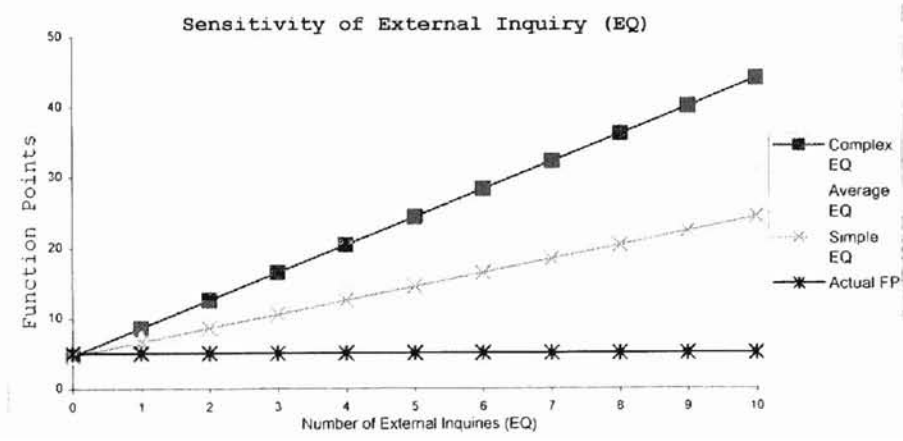
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE FPLAN PROGRAM

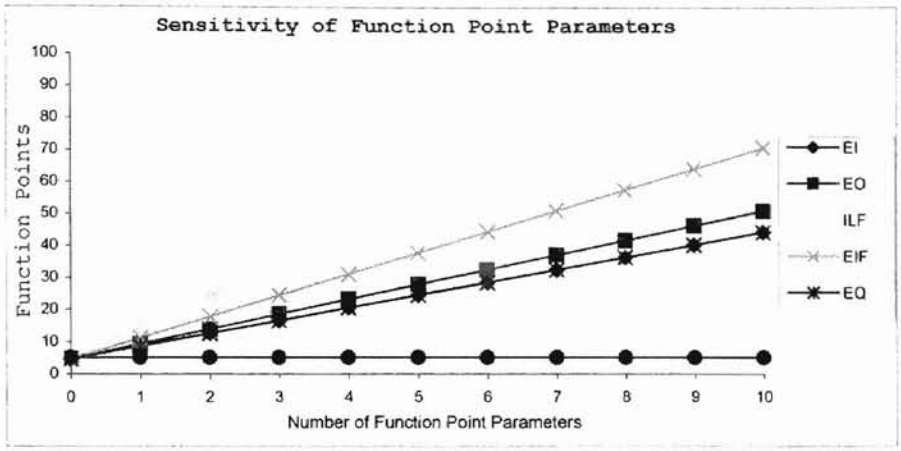


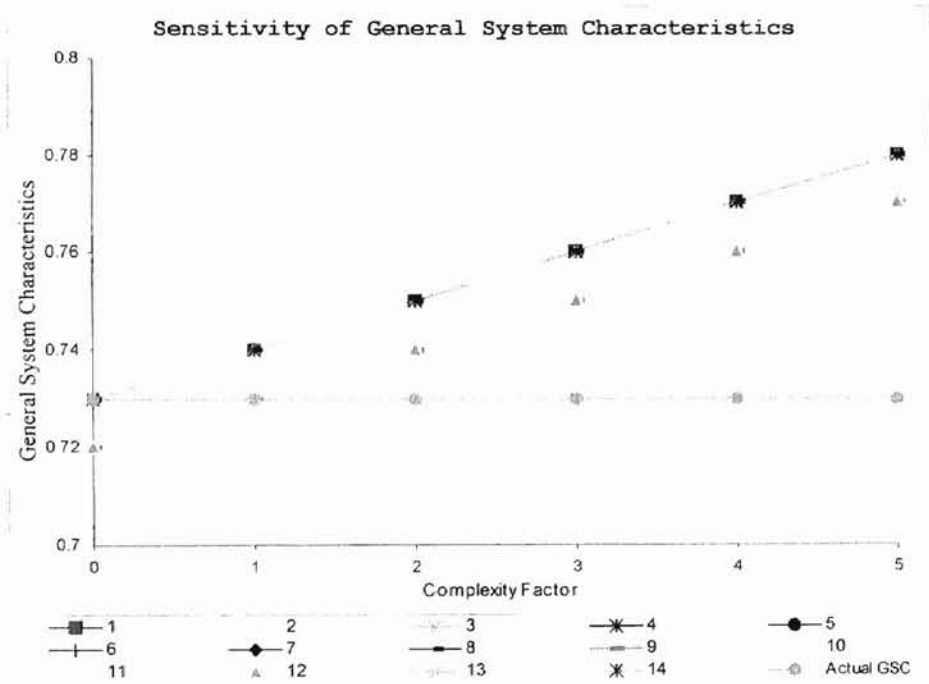



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

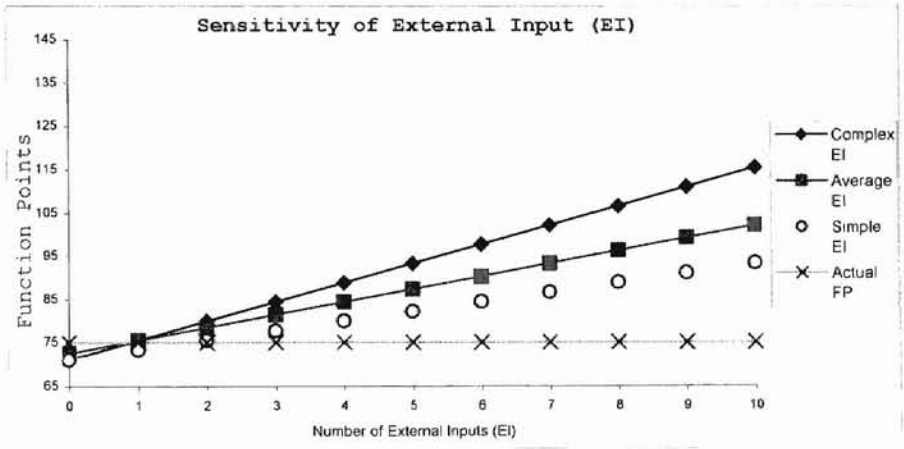

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

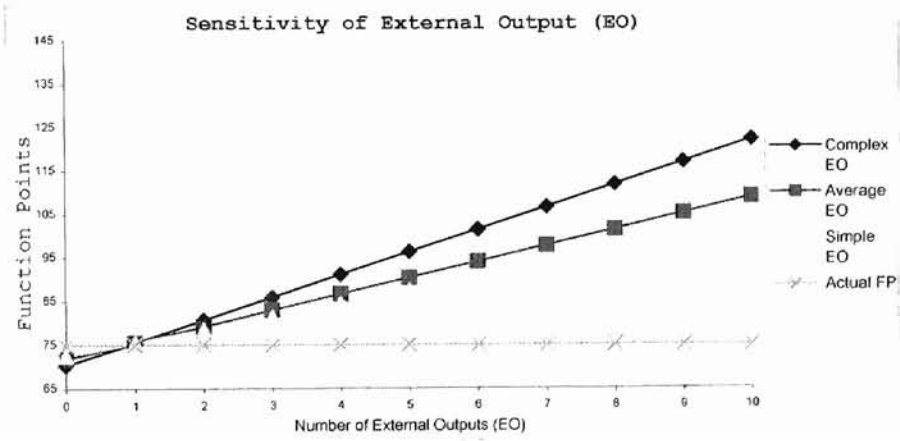

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquires (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

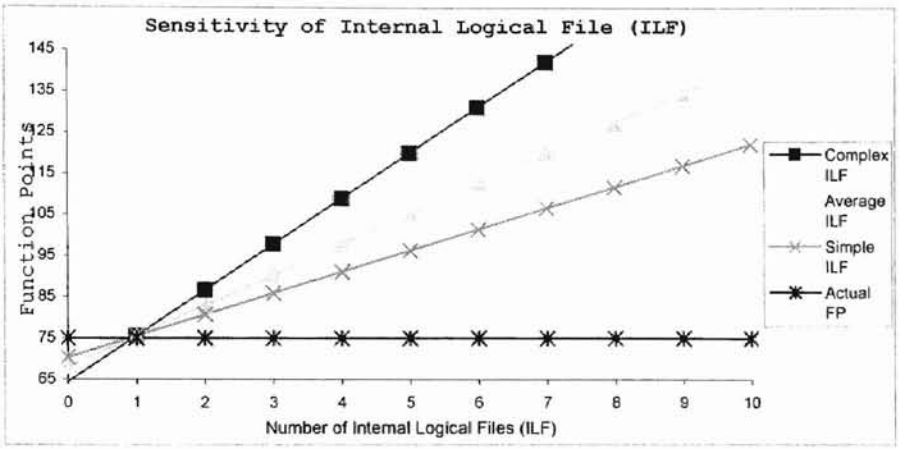

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ

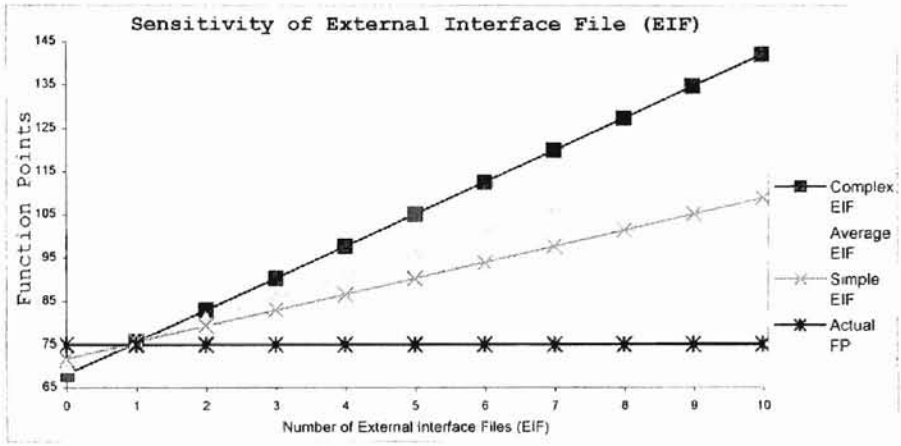

Sensitivity of General System Characteristics
General System Characteristics
0.8
0.78
0.76
0.74
0.72
0.7
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

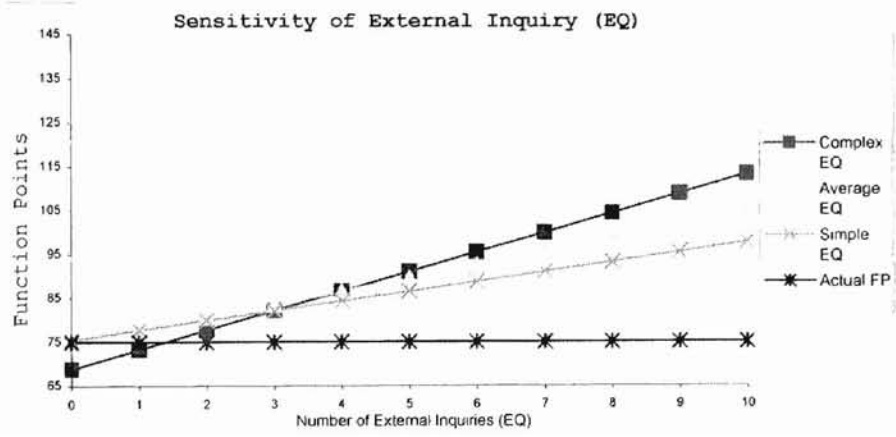
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE GHOSTVIEW PROGRAM

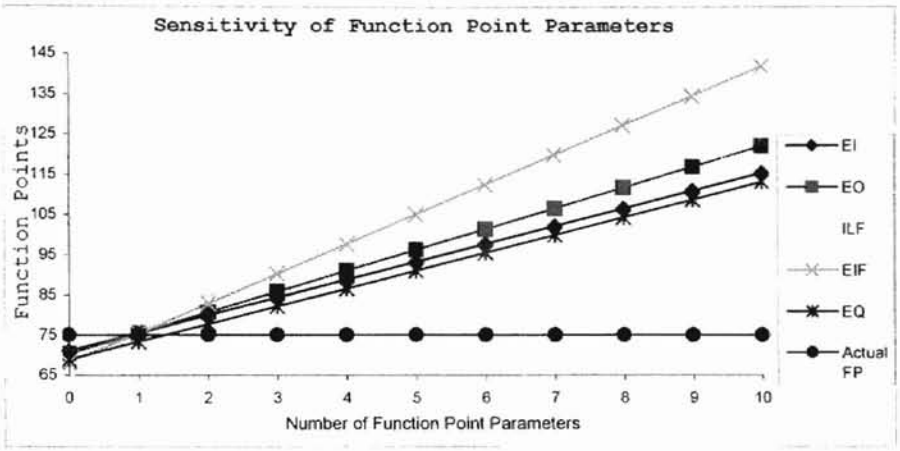


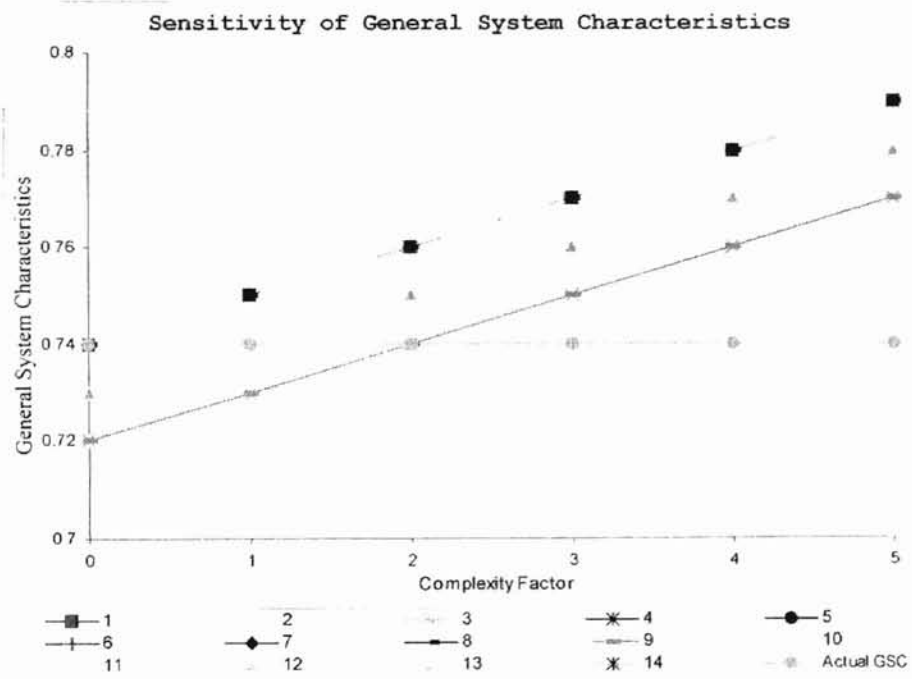



Sensitivity of Internal Logical File (ILF)
Function Points
145
135
125
115
105
95
85
75
65
0
1
2
3
4
5
6
7
8
9
10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

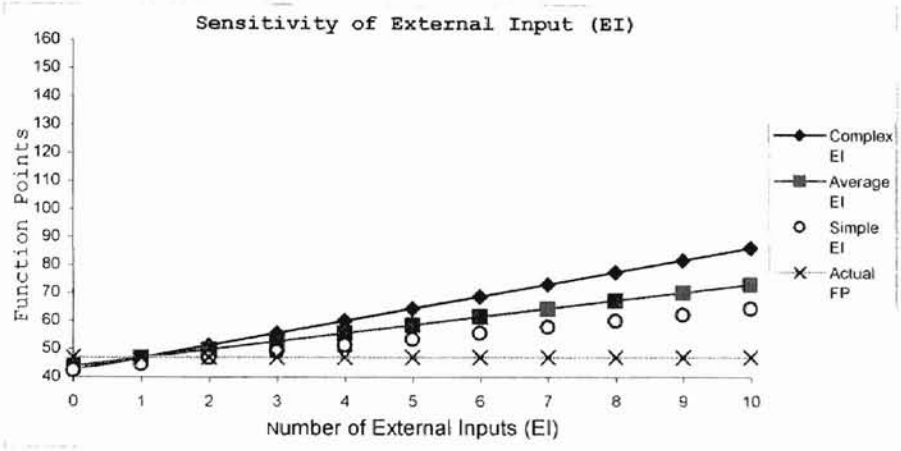

Sensitivity of External Interface File (EIF)
Function Points
145
135
125
115
105
95
85
75
65
0
1
2
3
4
5
6
7
8
9
10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

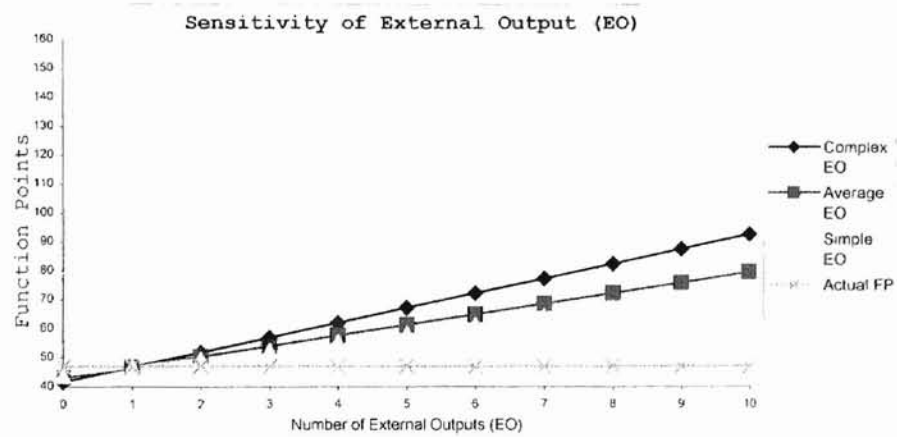

Sensitivity of External Inquiry (EQ)
Function Points
145
135
125
115
105
95
85
75
65
0
1
2
3
4
5
6
7
8
9
10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

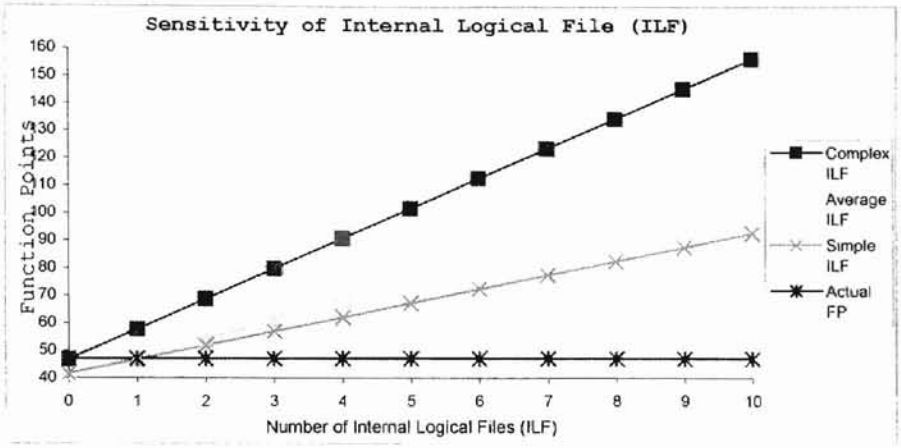

Sensitivity of Function Point Parameters
Function Points
145
135
125
115
105
95
85
75
65
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

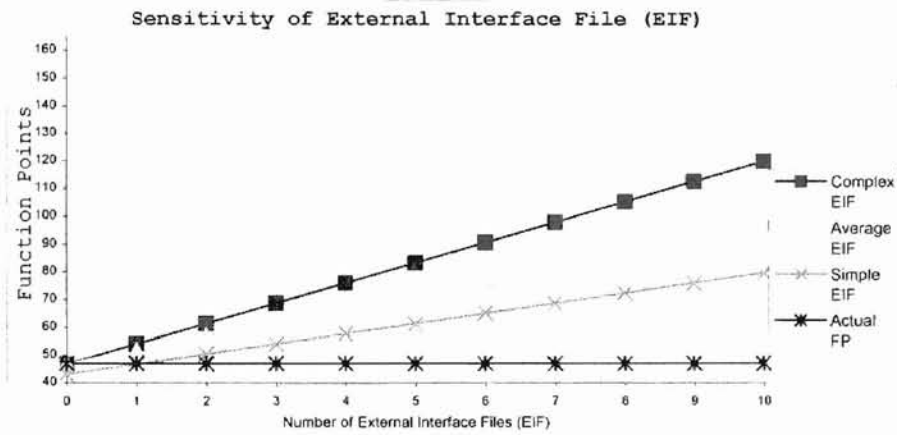

Sensitivity of General System Characteristics
General System Characteristics
0.8
0.78
0.76
0.74
0.72
0.7
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

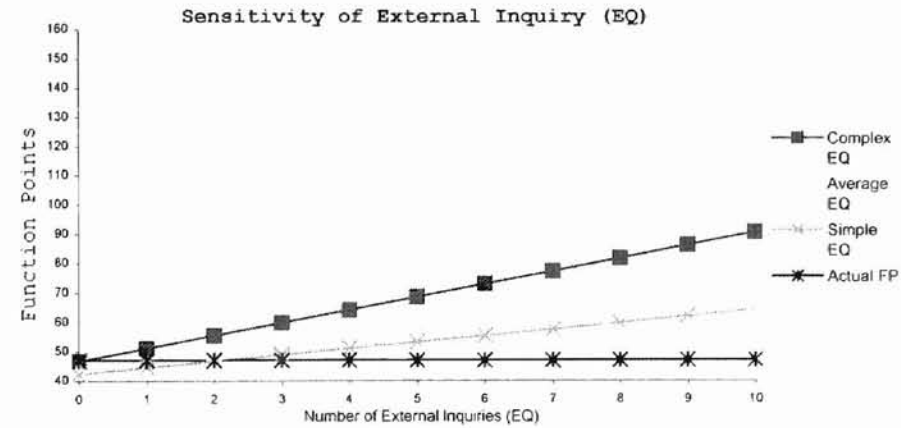
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE GZIP PROGRAM

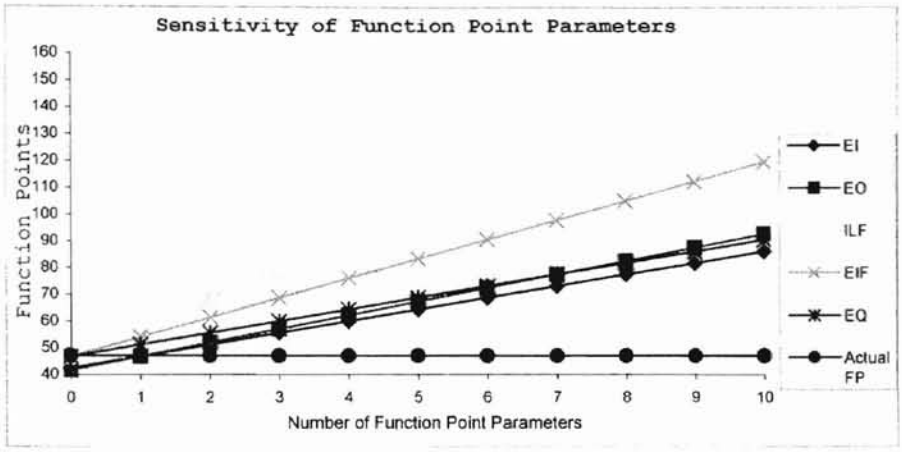


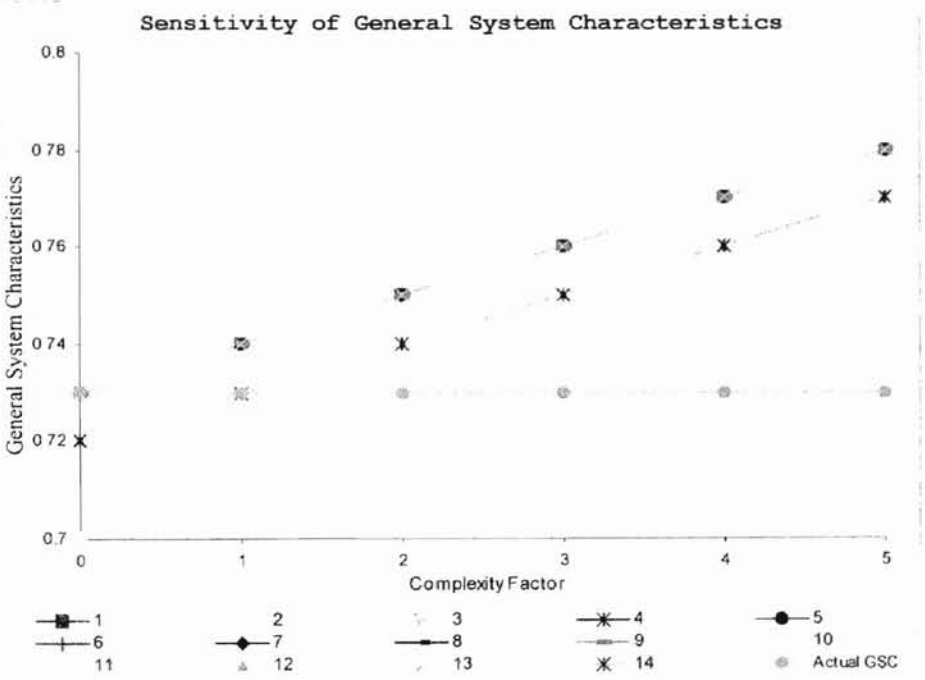



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

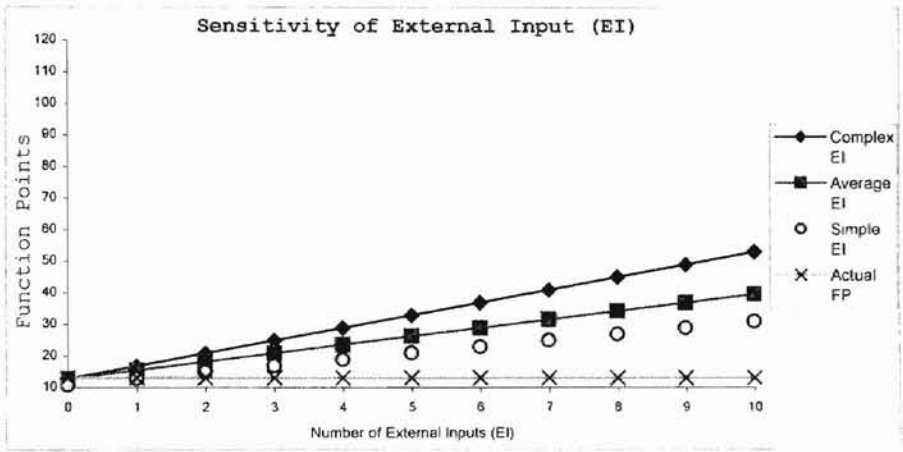

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

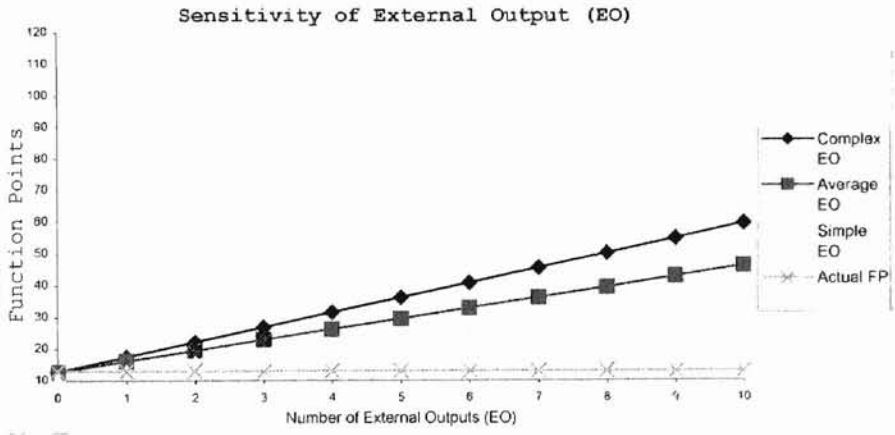

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

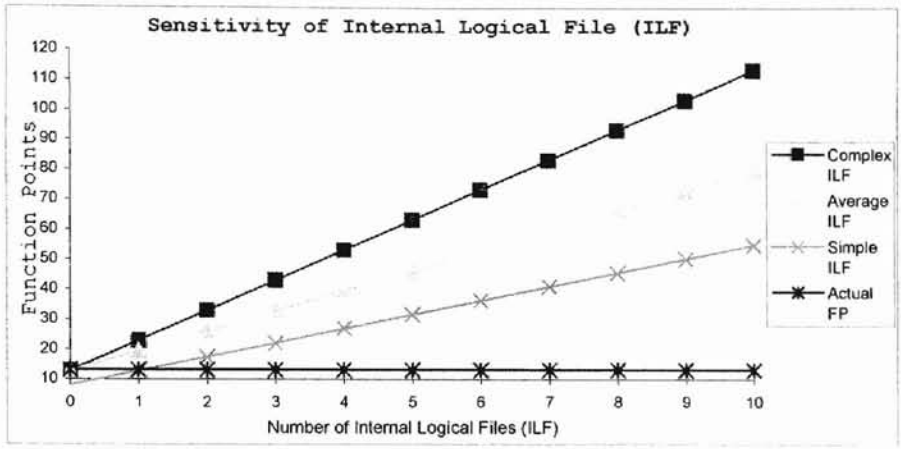

Sensitivity of Function Point Parameters
Function Points
160
150
140
130
120
110
100
90
80
70
60
50
40
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

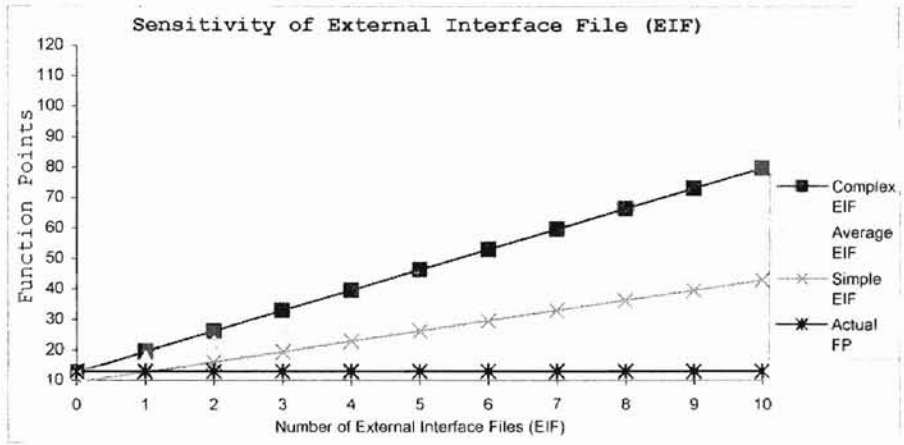

Sensitivity of General System Characteristics
General System Characteristics
0.8
0.78
0.76
0.74
0.72
0.7
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

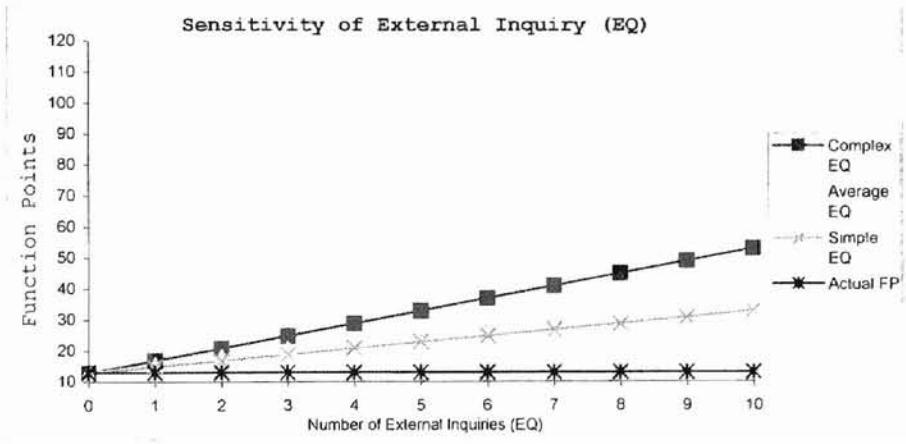
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE IMPORT PROGRAM

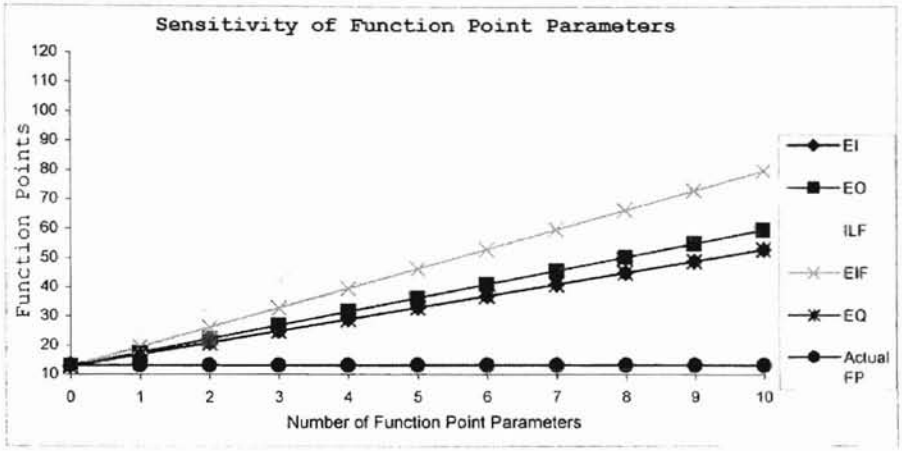


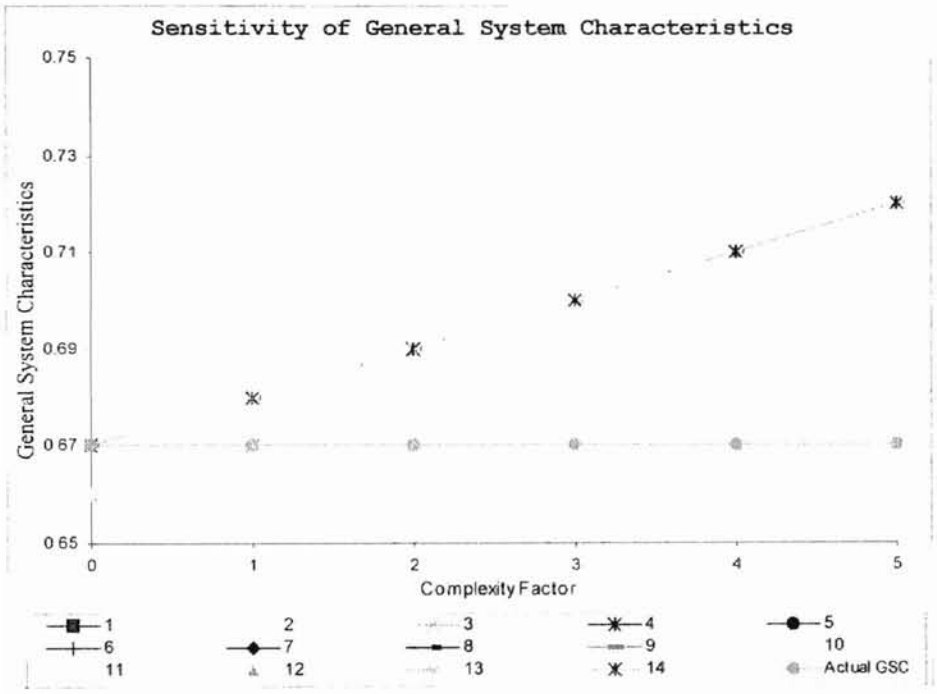



Sensitivity of Internal Logical File (ILF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

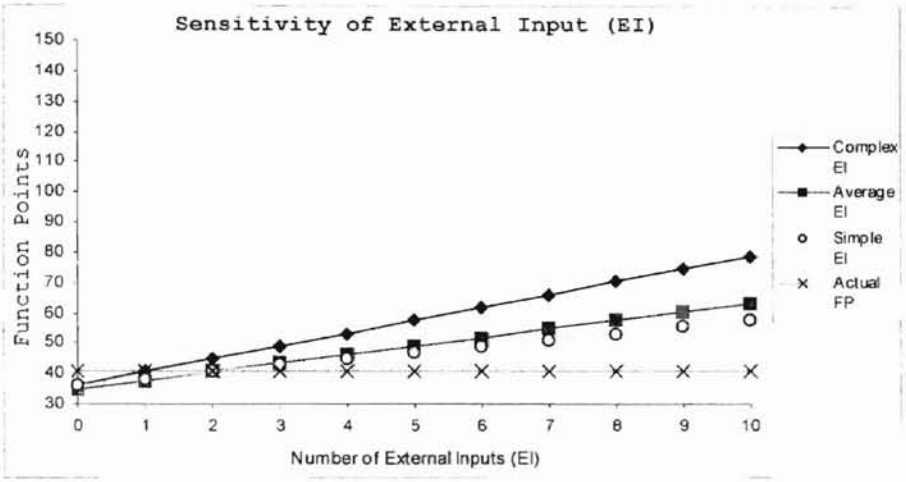

Sensitivity of External Interface File (EIF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

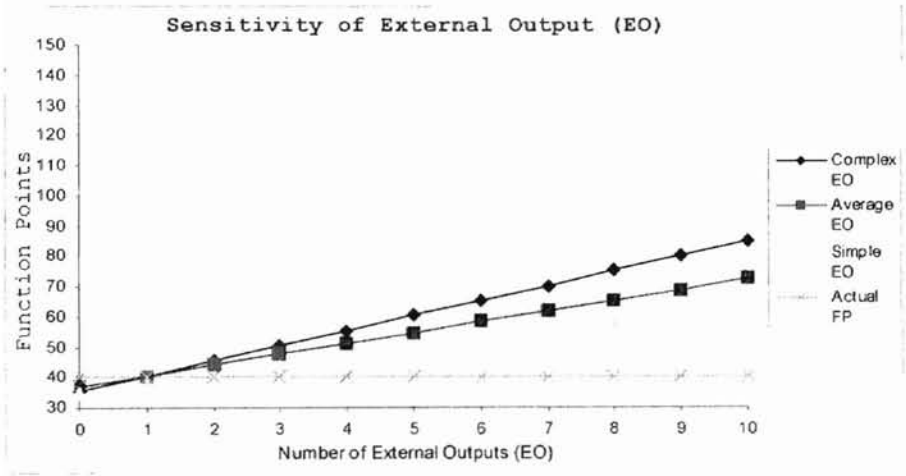

Sensitivity of External Inquiry (EQ)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

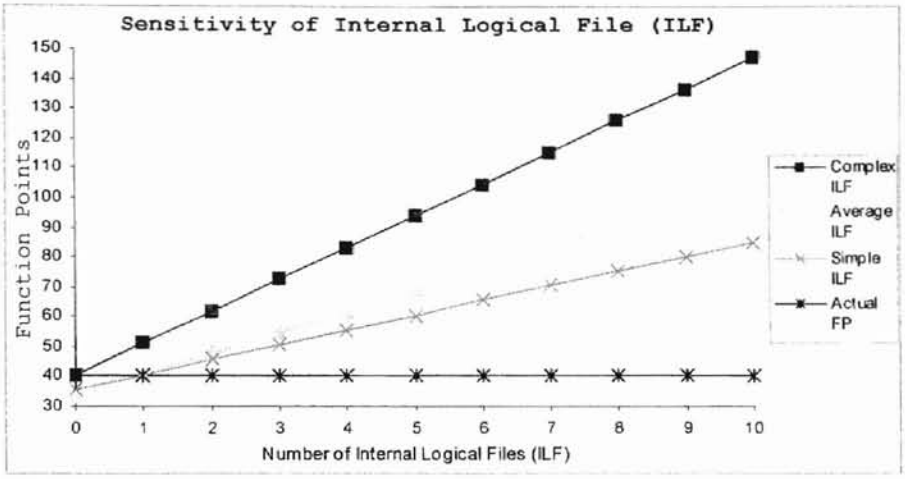

Sensitivity of Function Point Parameters
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

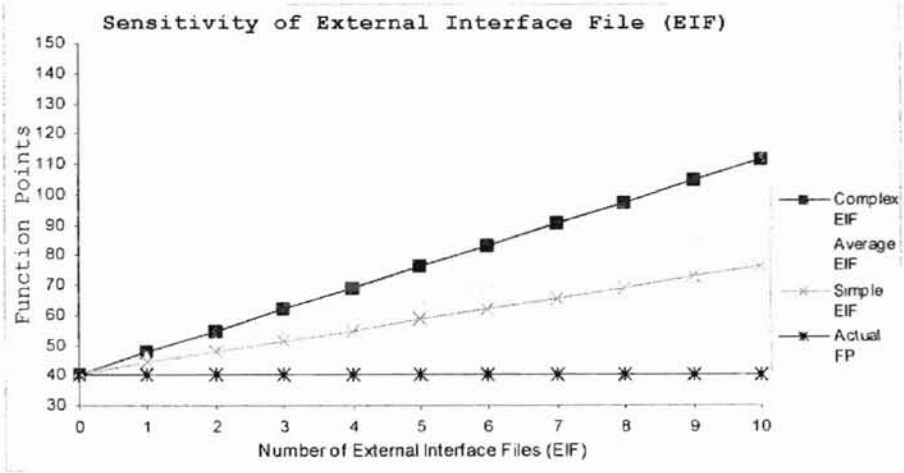

Sensitivity of General System Characteristics
General System Characteristics
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

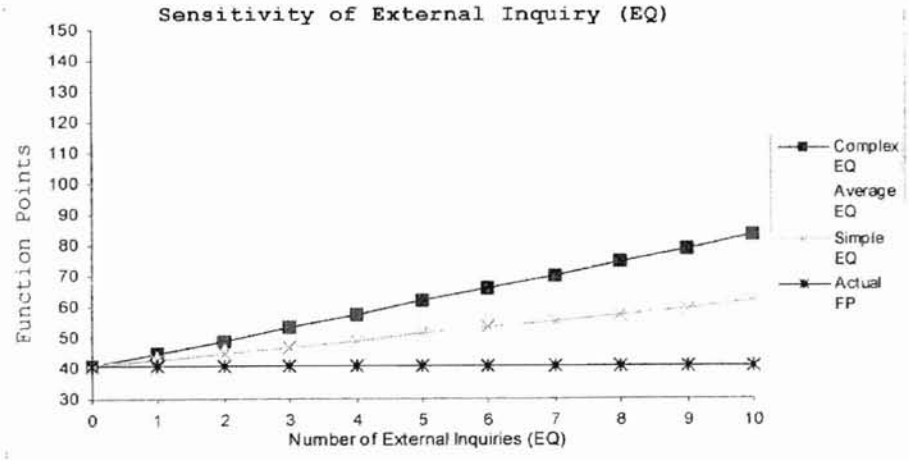
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE INDENT PROGRAM

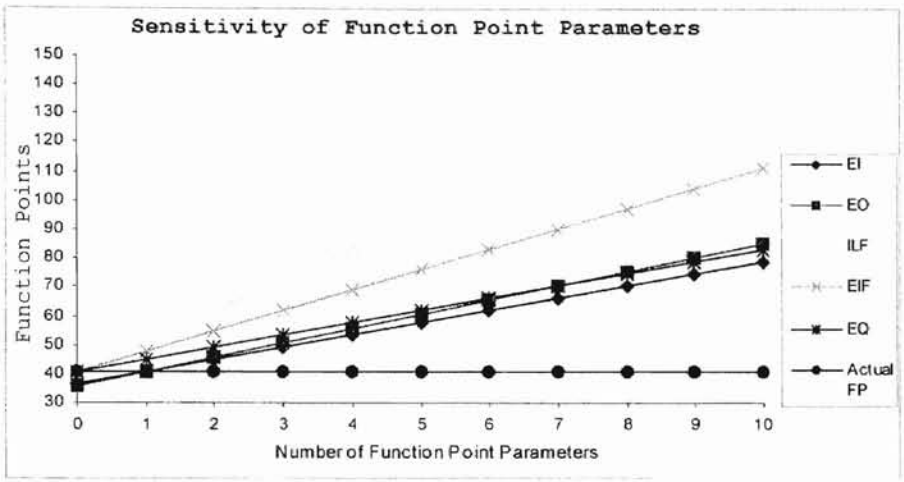


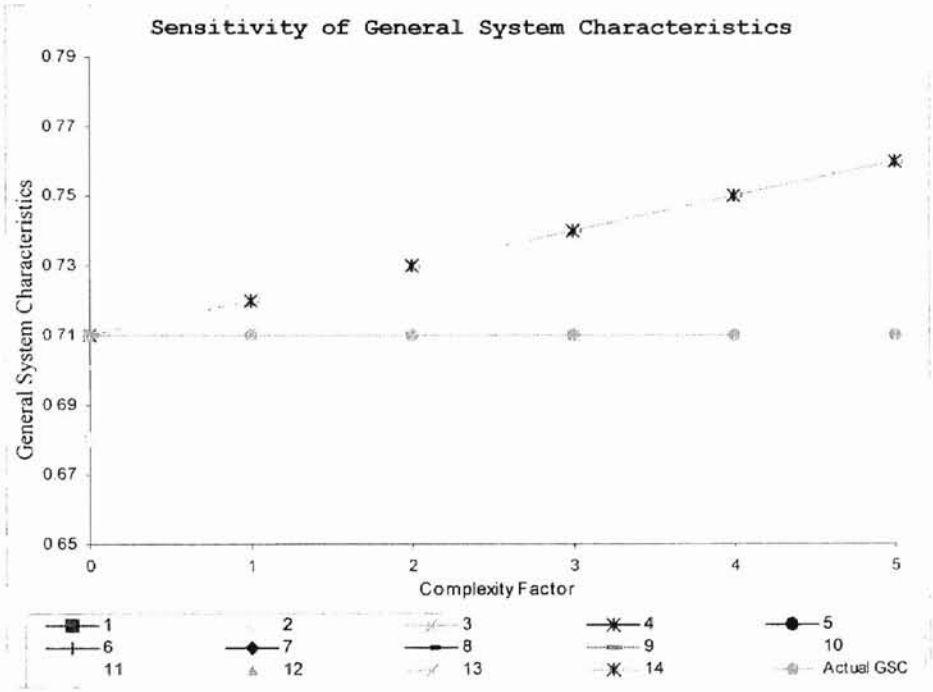



Sensitivity of Internal Logical File (ILF)
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

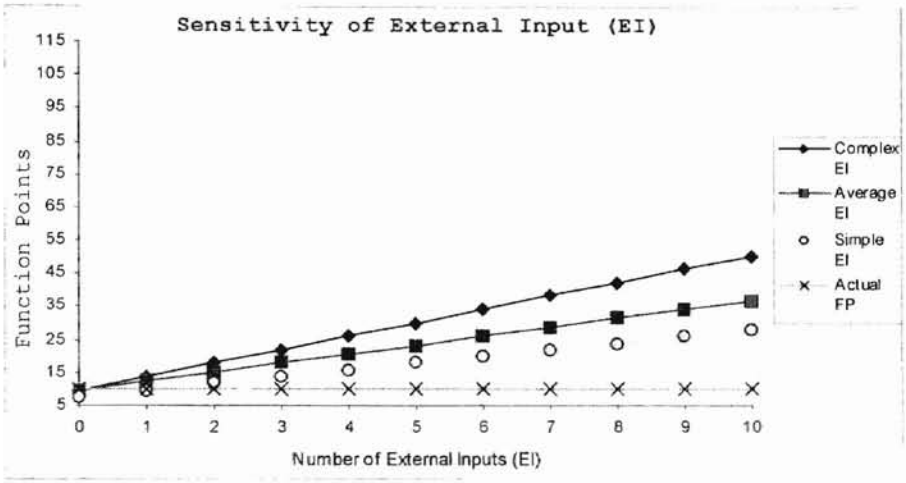

Sensitivity of External Interface File (EIF)
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

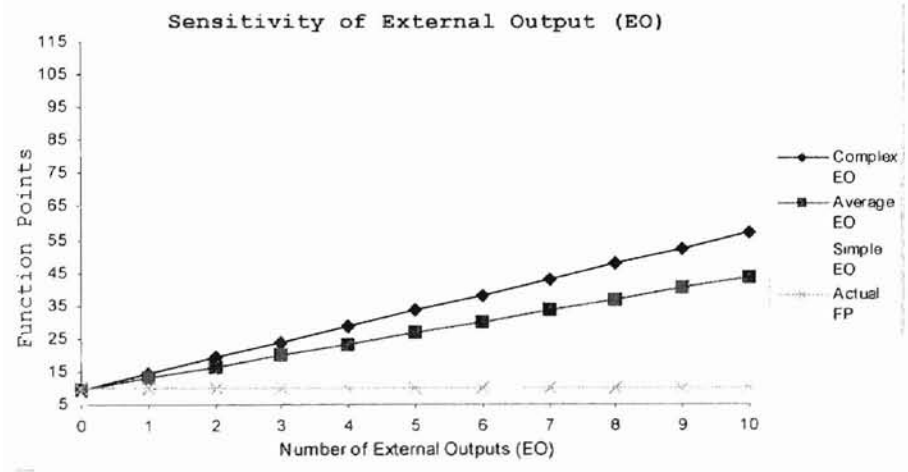

Sensitivity of External Inquiry (EQ)
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

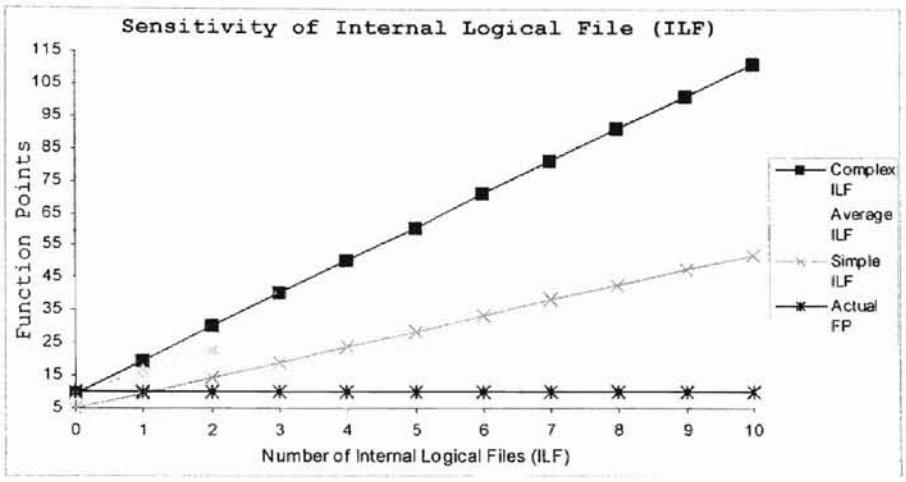

Sensitivity of Function Point Parameters
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

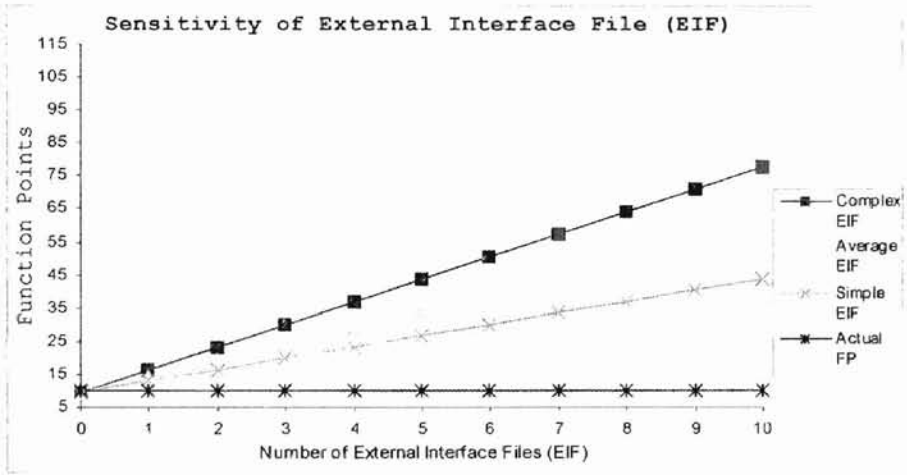

Sensitivity of General System Characteristics
General System Characteristics
0.79
0.77
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

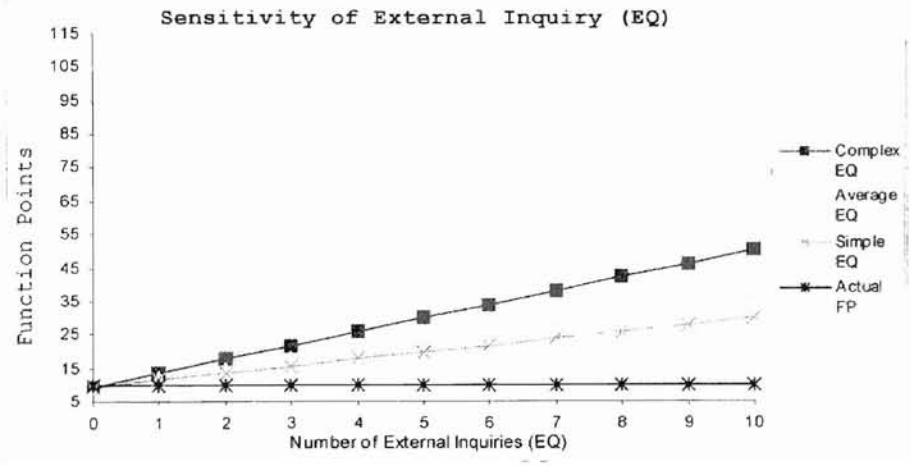
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE LANDER PROGRAM

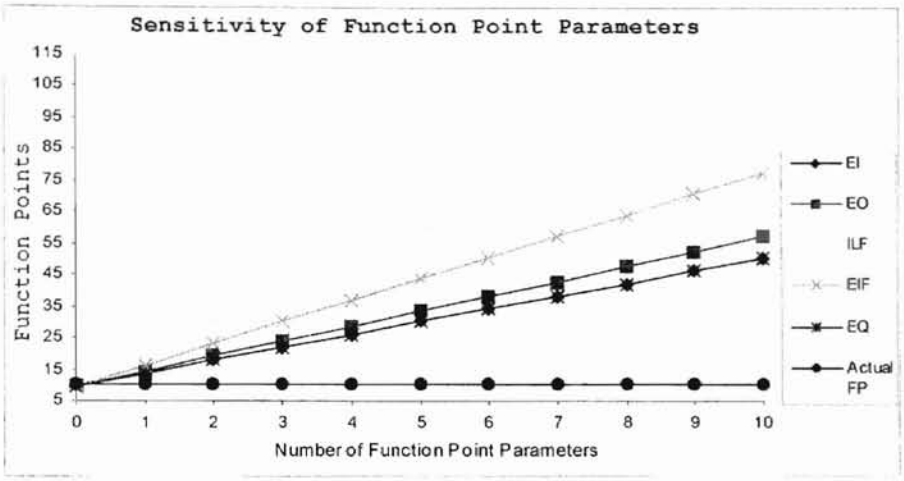


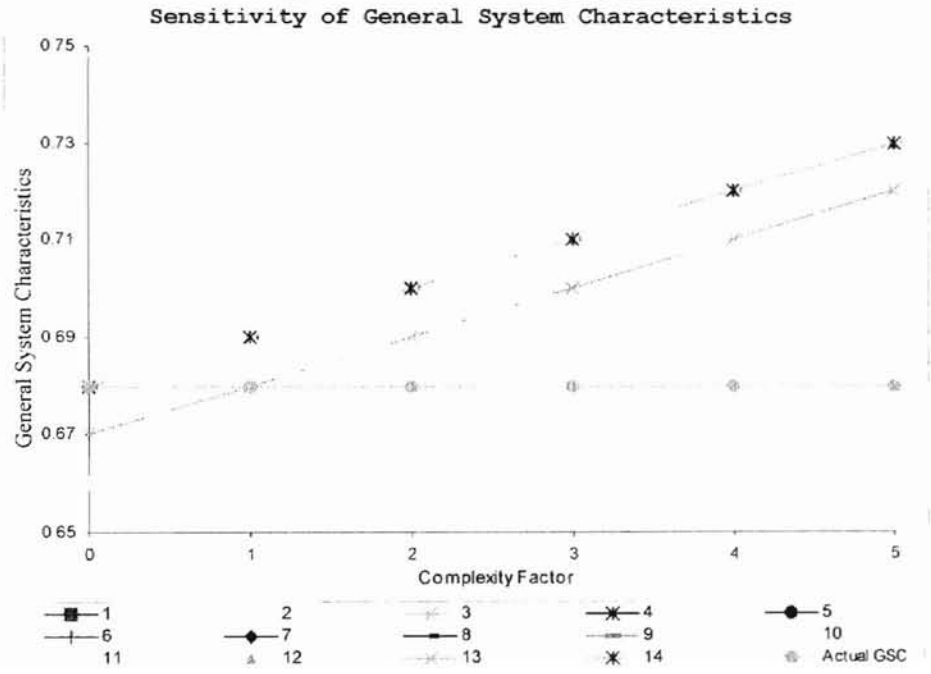



Sensitivity of Internal Logical File (ILF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

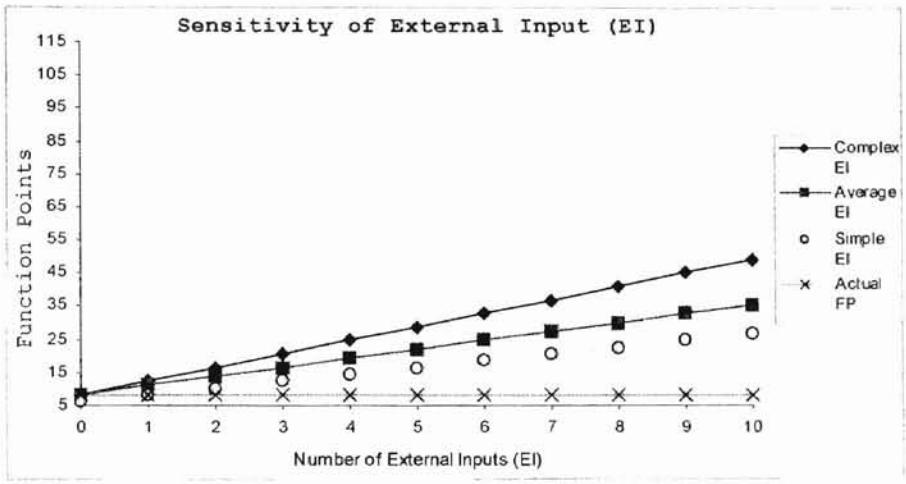

Sensitivity of External Interface File (EIF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

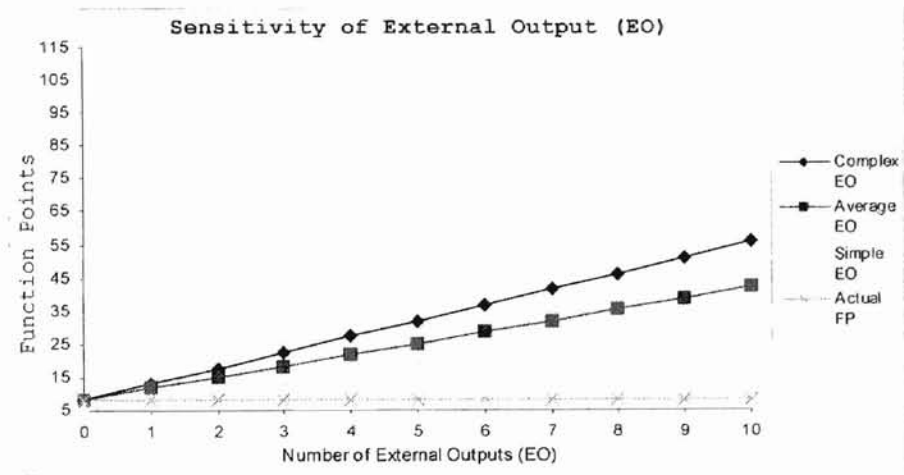

Sensitivity of External Inquiry (EQ)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

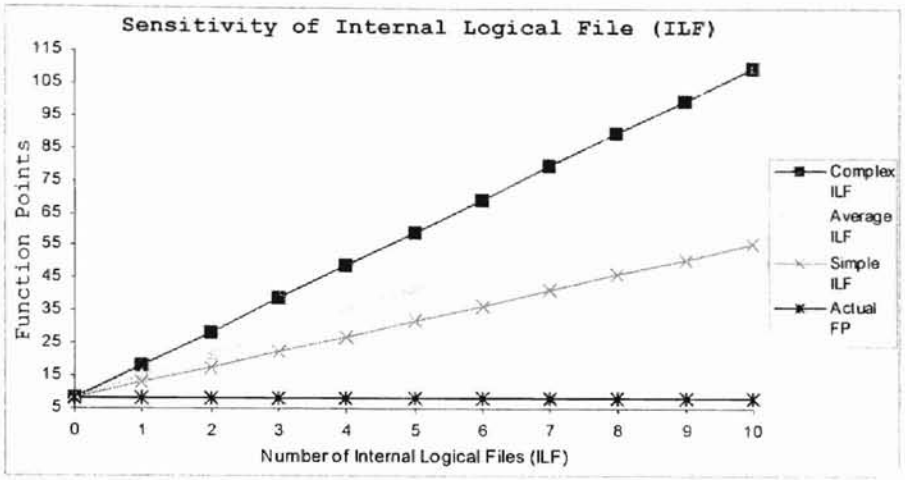

Sensitivity of Function Point Parameters
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

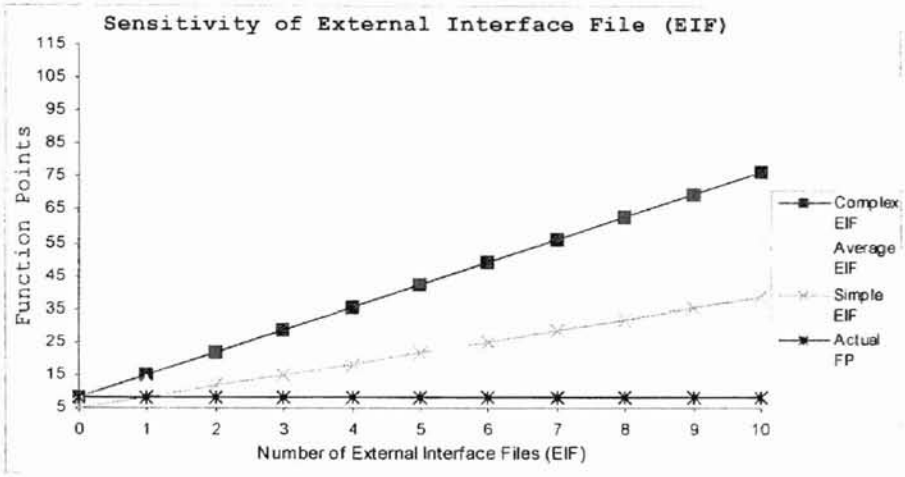

Sensitivity of General System Characteristics
General System Characteristics
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

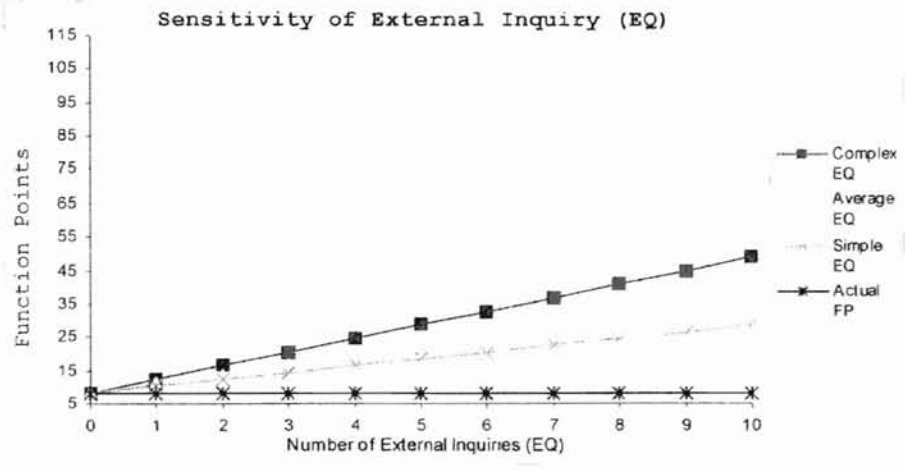
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE MAN PROGRAM

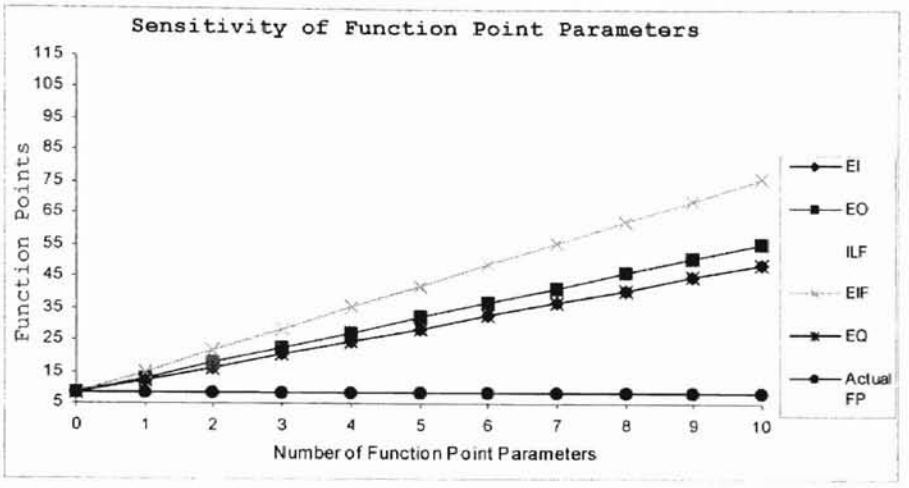


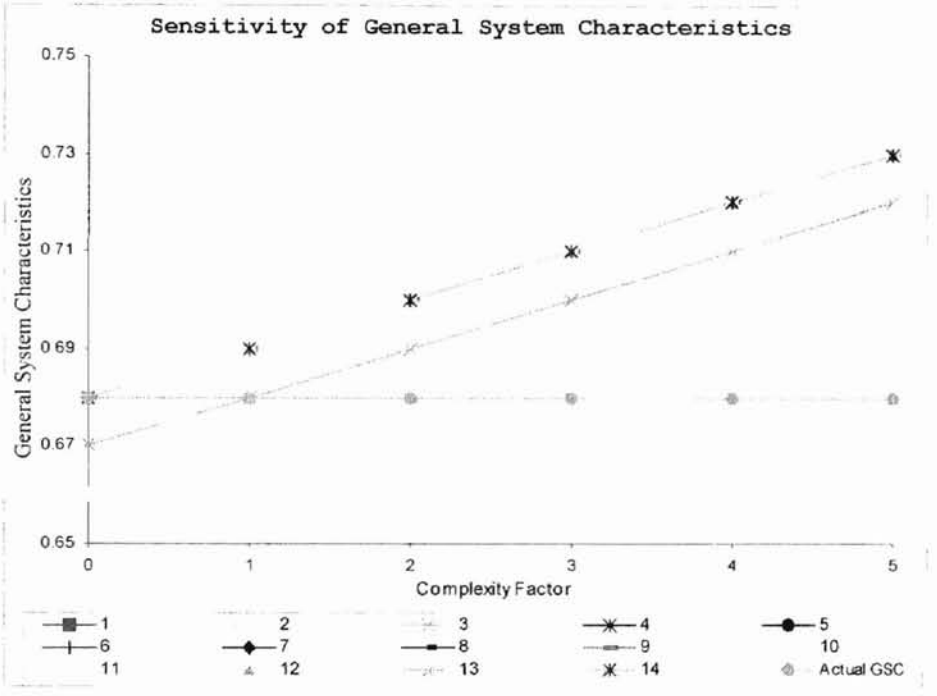



Sensitivity of Internal Logical File (ILF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

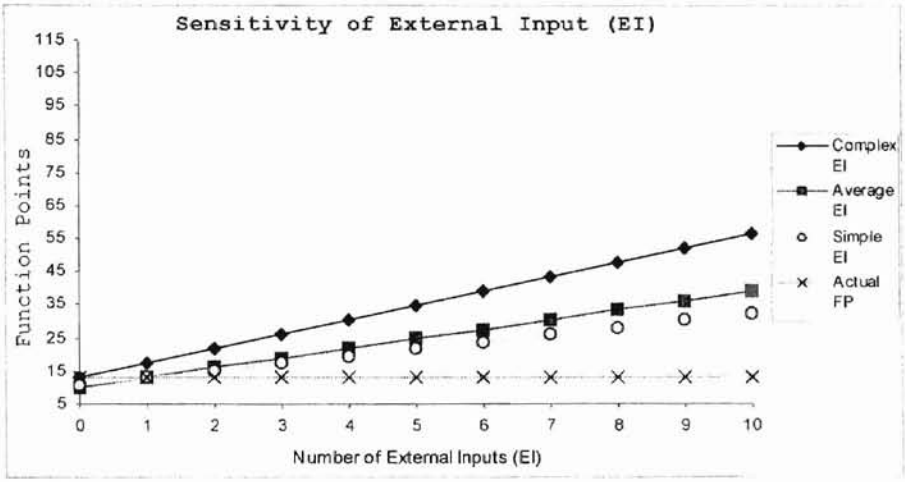

Sensitivity of External Interface File (EIF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

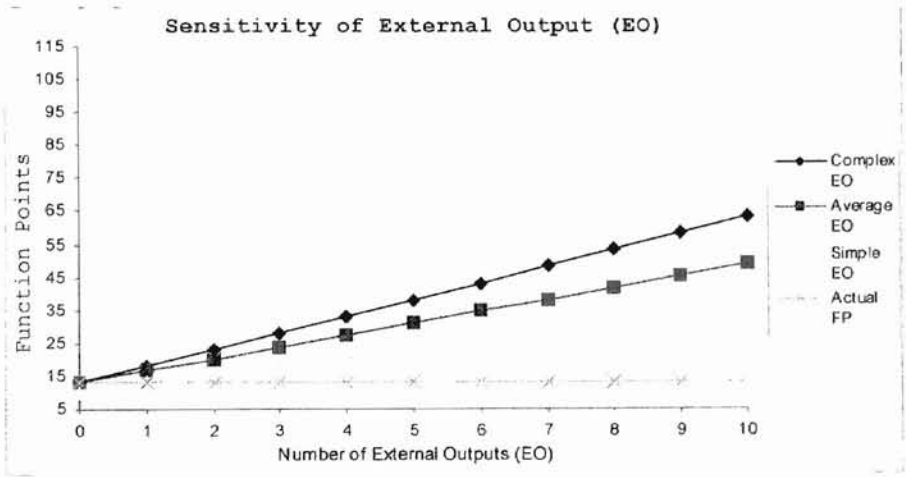

Sensitivity of External Inquiry (EQ)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

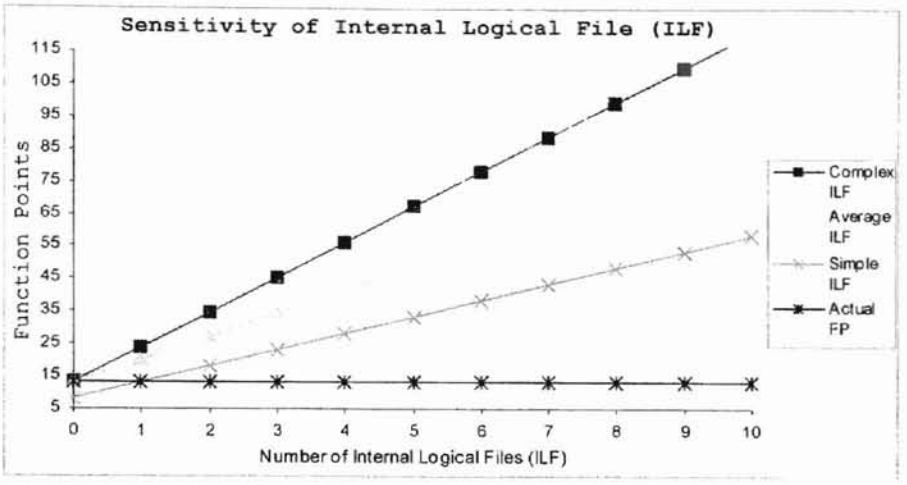

Sensitivity of Function Point Parameters
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

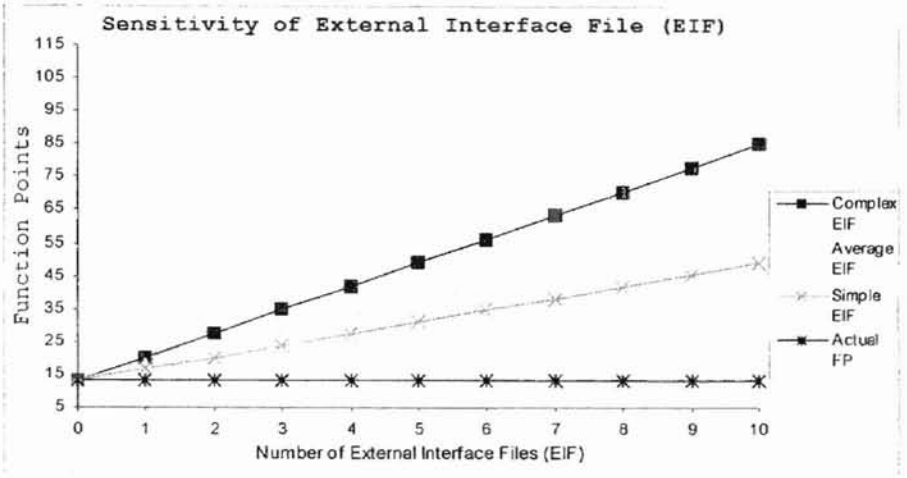

Sensitivity of General System Characteristics
General System Characteristics
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

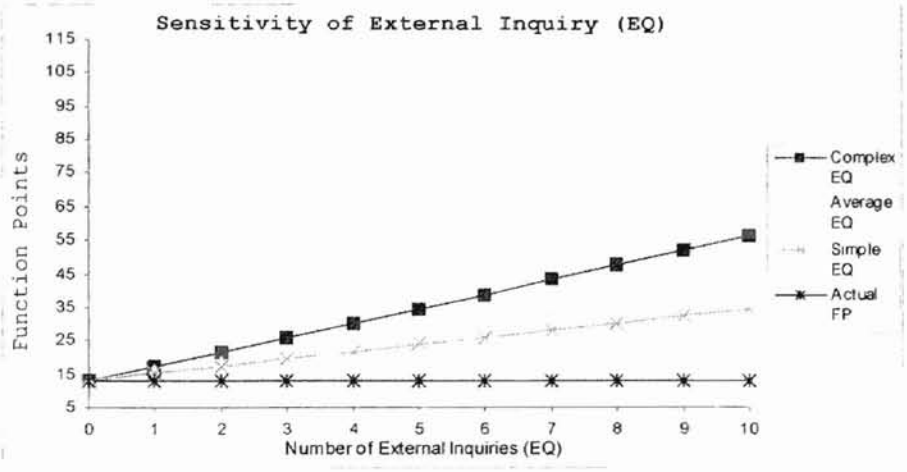
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE MOGRIFY PROGRAM

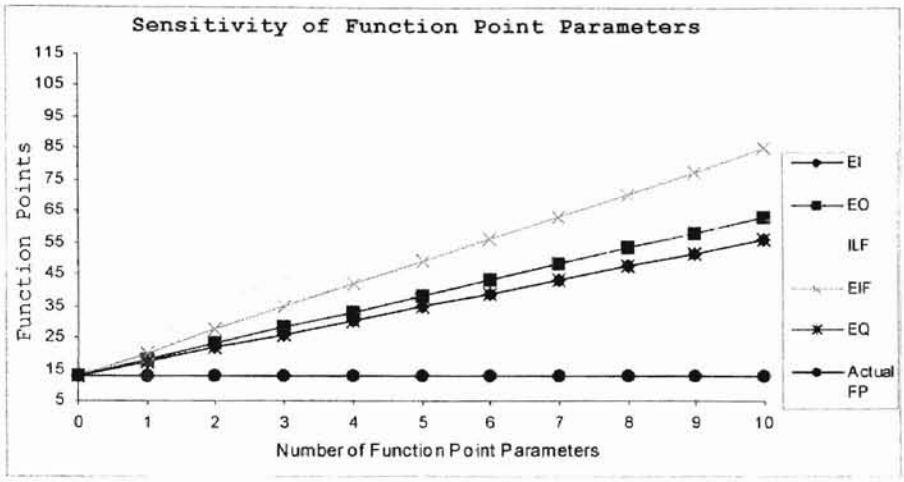


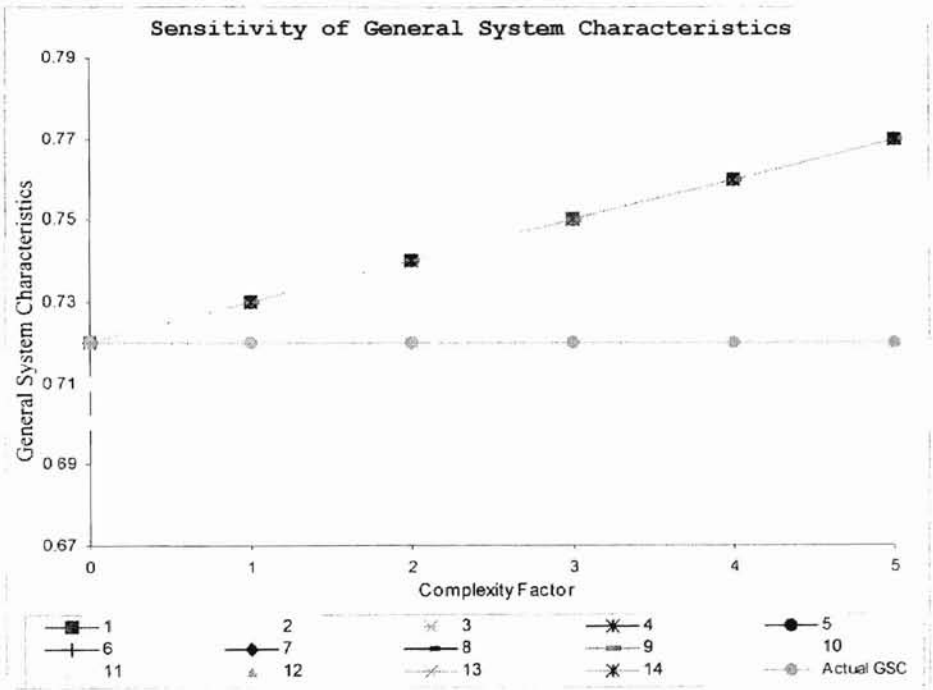



Sensitivity of Internal Logical File (ILF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

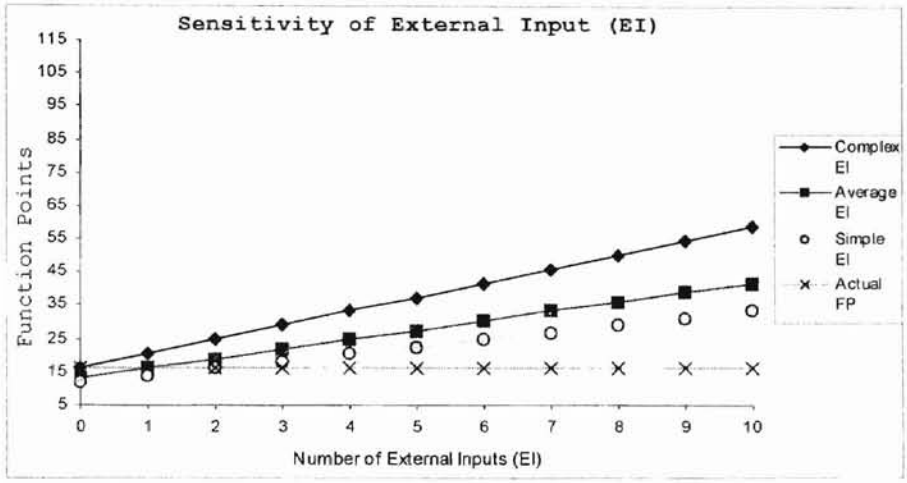

Sensitivity of External Interface File (EIF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

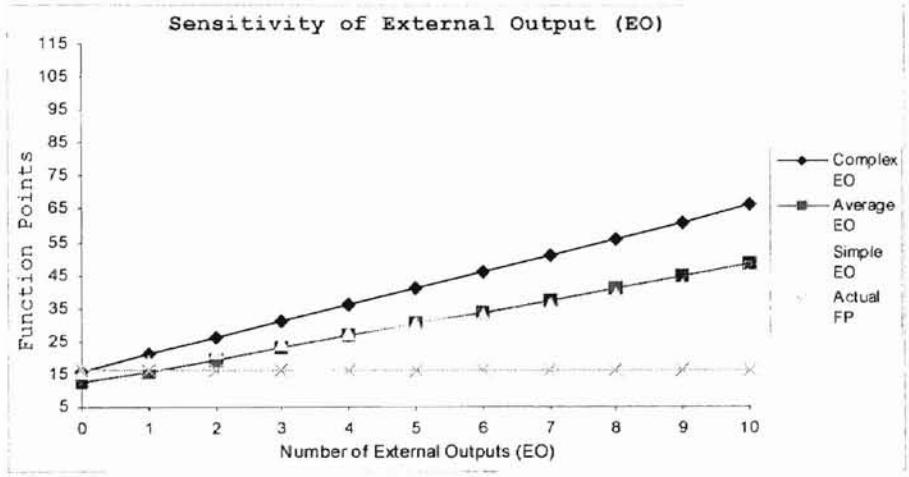

Sensitivity of External Inquiry (EQ)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

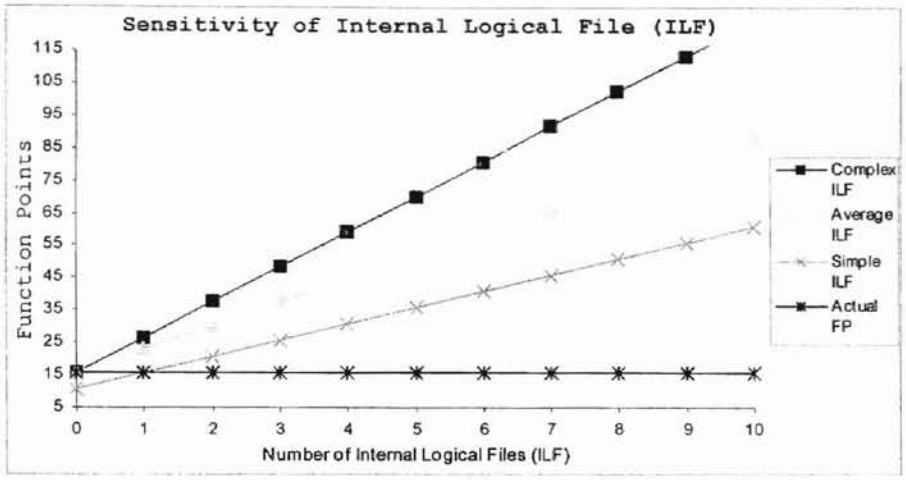

Sensitivity of Function Point Parameters
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

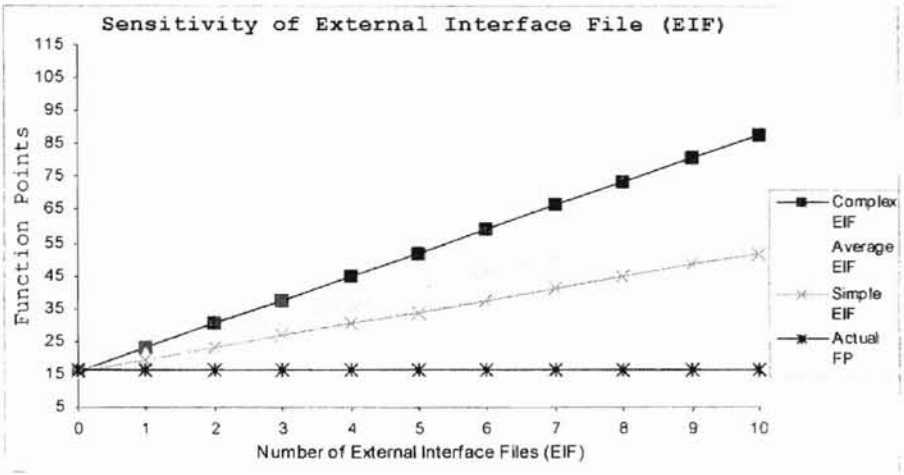

Sensitivity of General System Characteristics
General System Characteristics
0.79
0.77
0.75
0.73
0.71
0.69
0.67
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

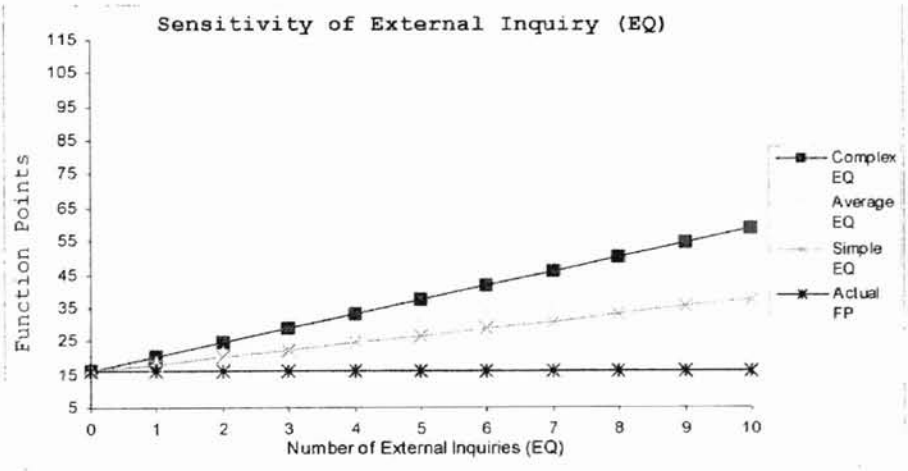
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE MONTAGE PROGRAM

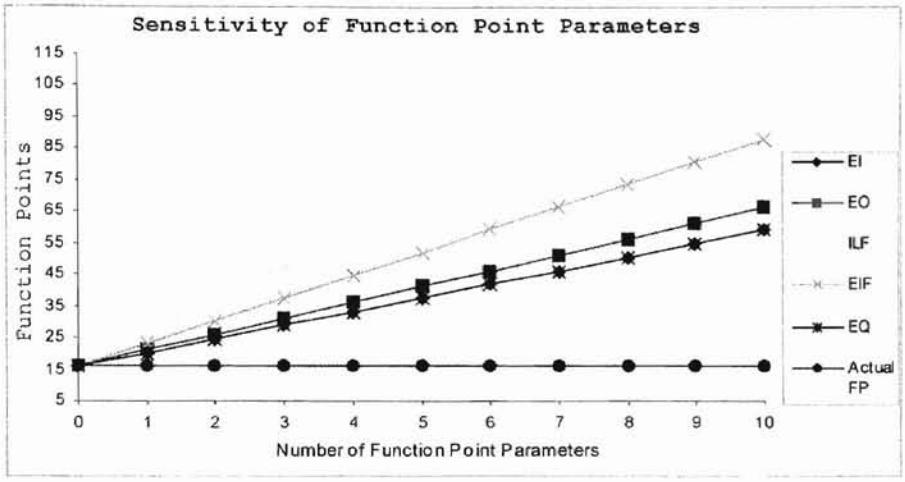


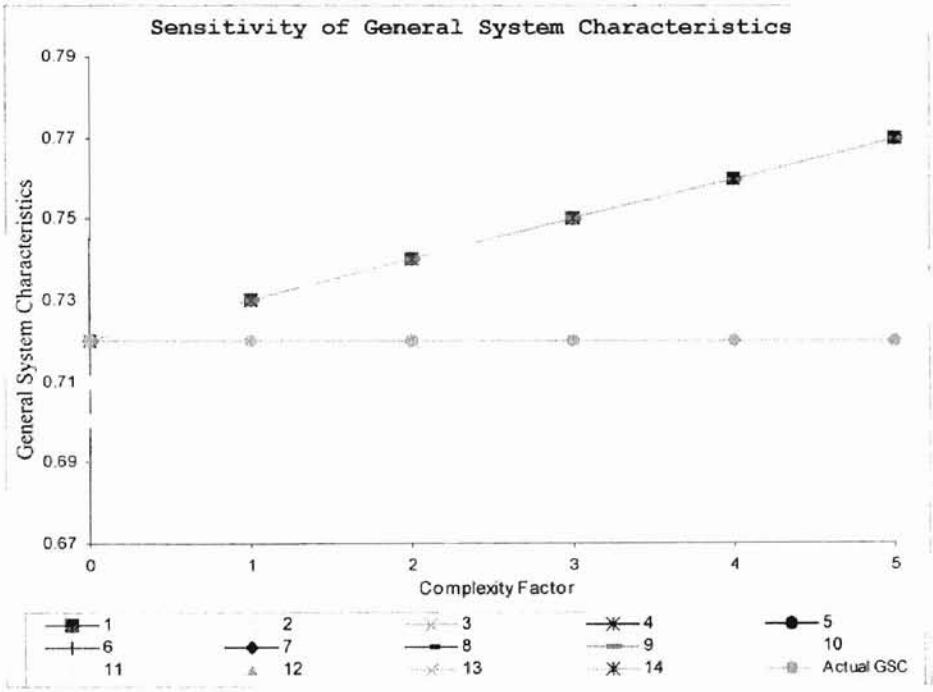



Sensitivity of Internal Logical File (ILF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

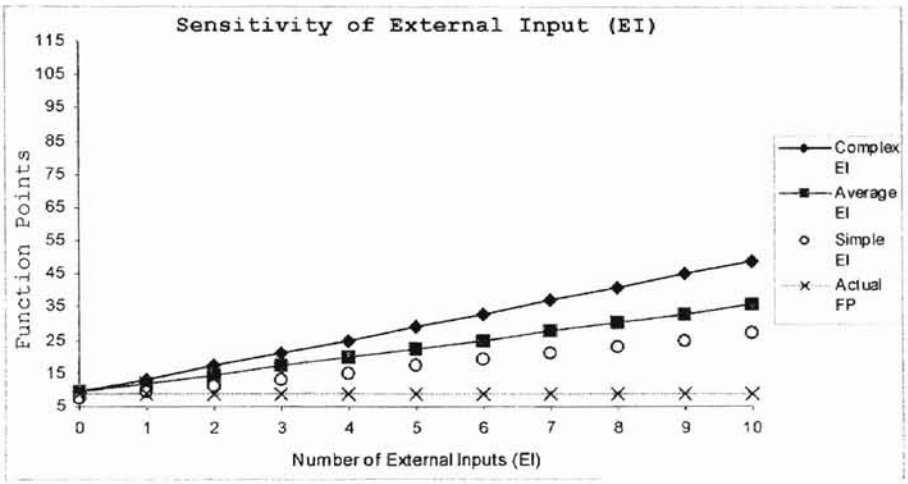

Sensitivity of External Interface File (EIF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

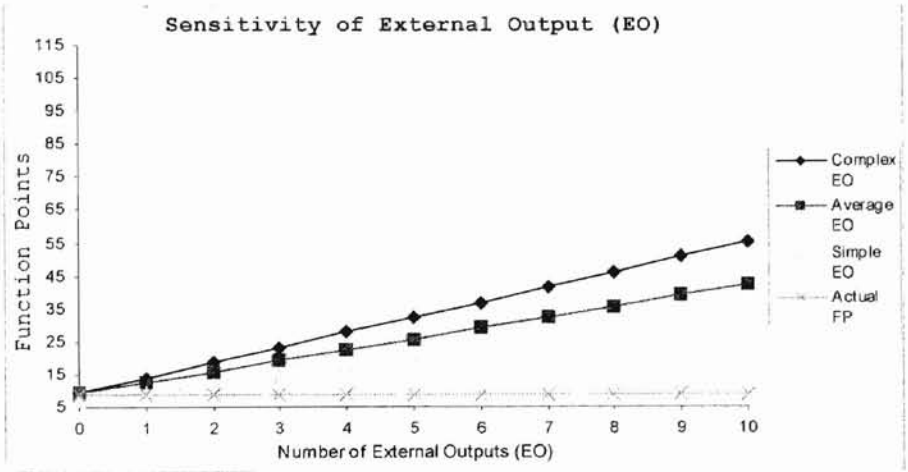

Sensitivity of External Inquiry (EQ)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

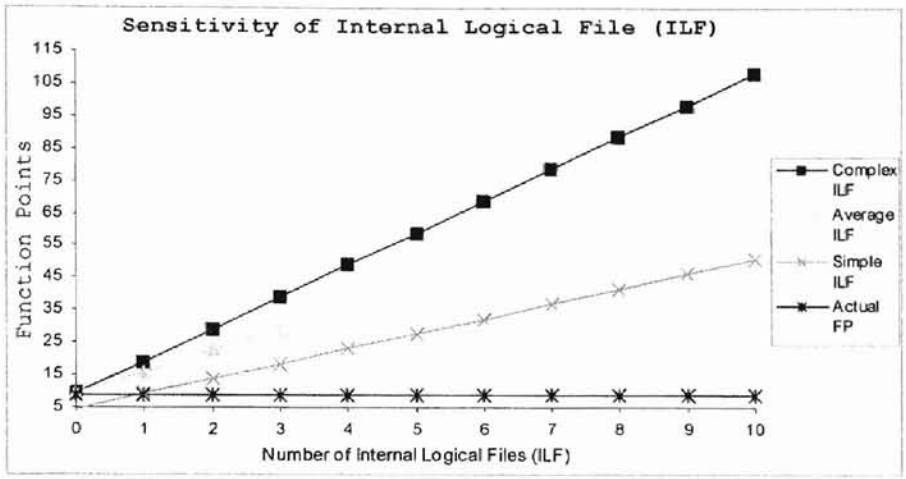

Sensitivity of Function Point Parameters
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

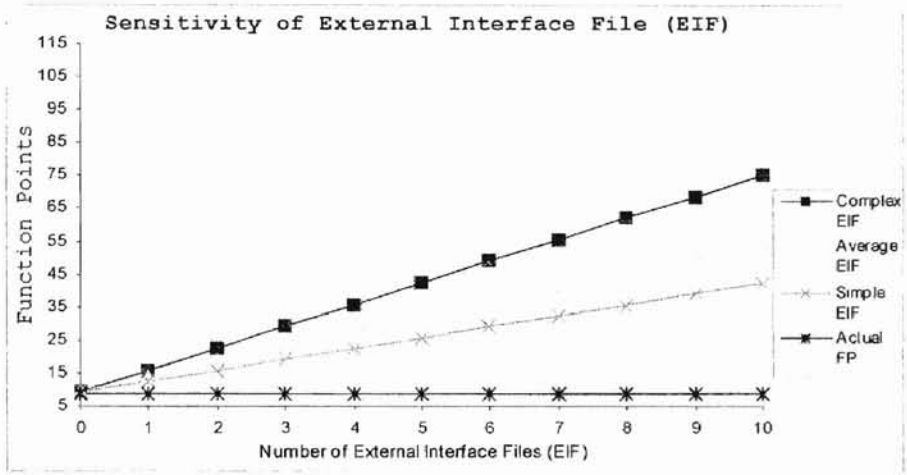

Sensitivity of General System Characteristics
General System Characteristics
0.79
0.77
0.75
0.73
0.71
0.69
0.67
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

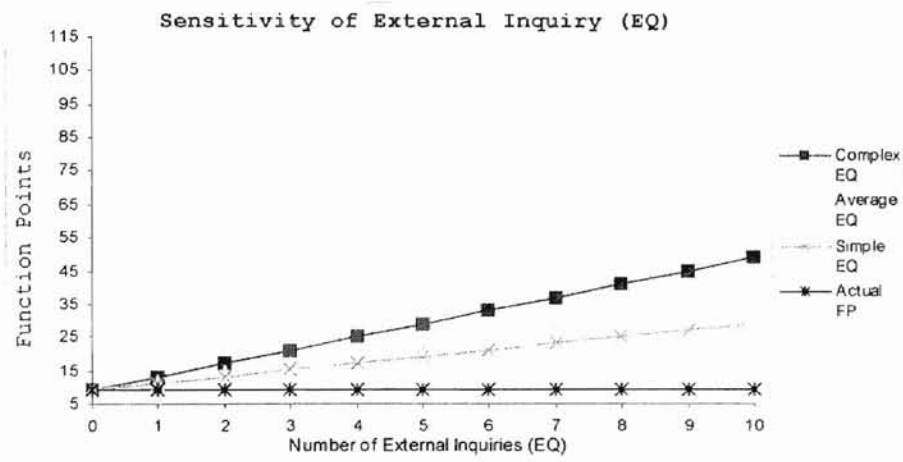
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE OTHELLO PROGRAM

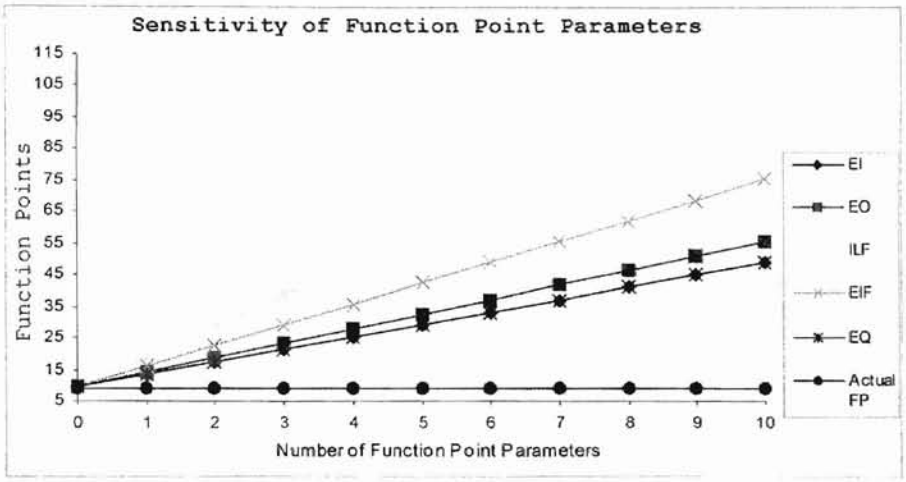


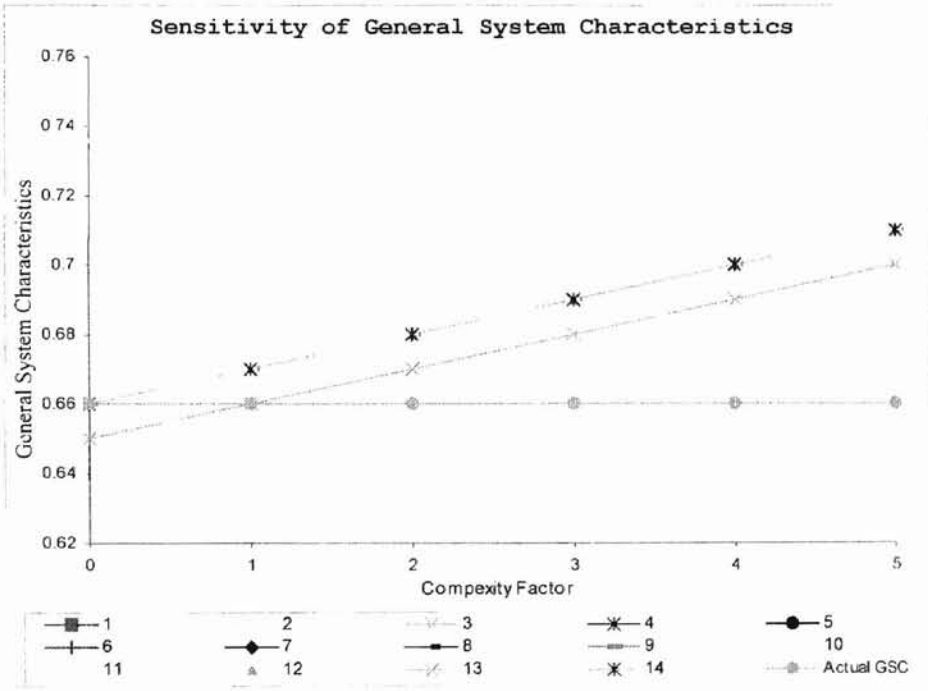



Sensitivity of Internal Logical File (ILF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

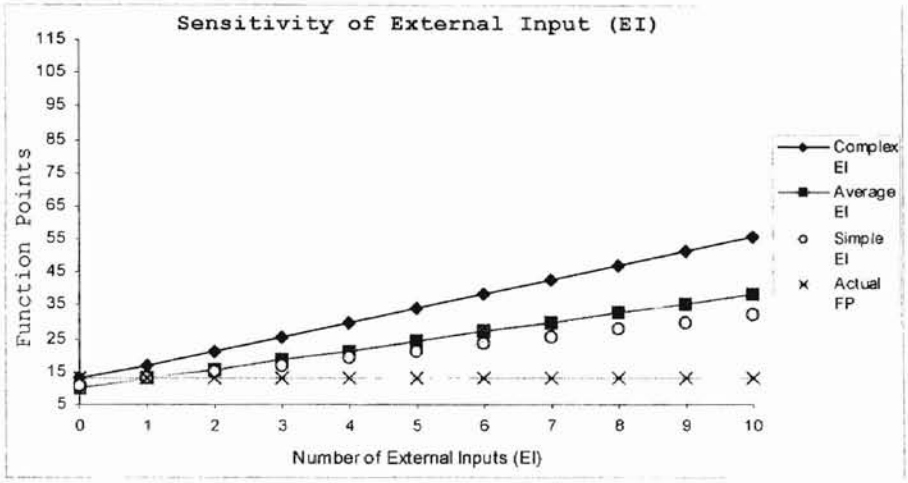

Sensitivity of External Interface File (EIF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

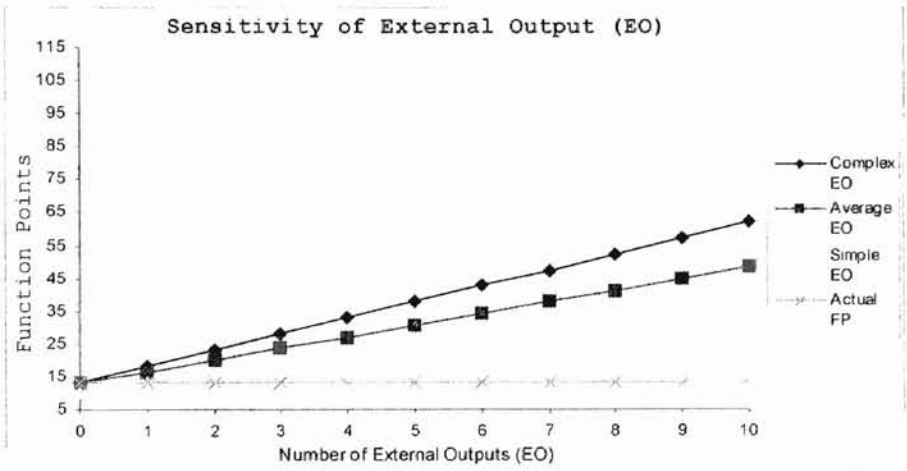

Sensitivity of External Inquiry (EQ)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

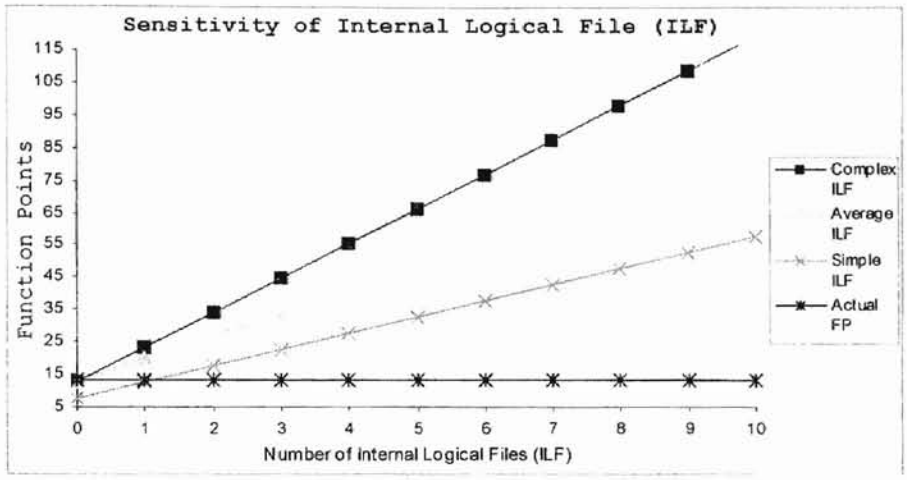

Sensitivity of Function Point Parameters
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

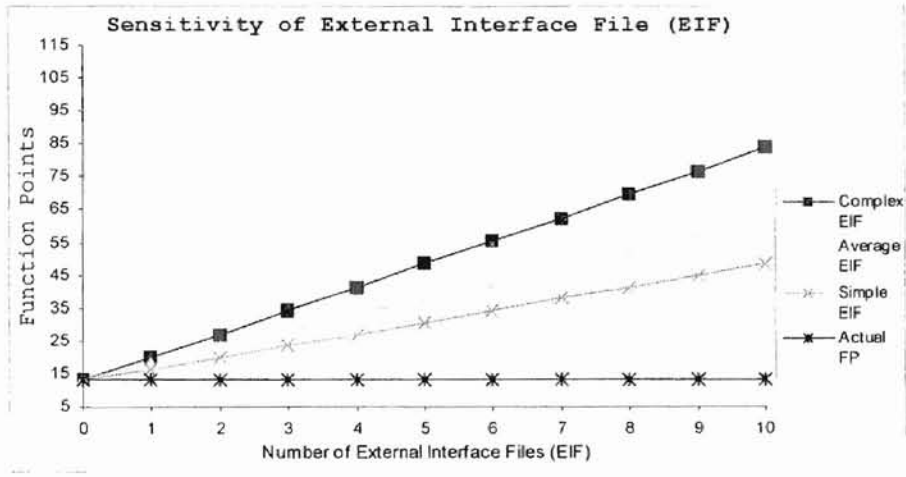

Sensitivity of General System Characteristics
General System Characteristics
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0
1
2
3
4
5
Compexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

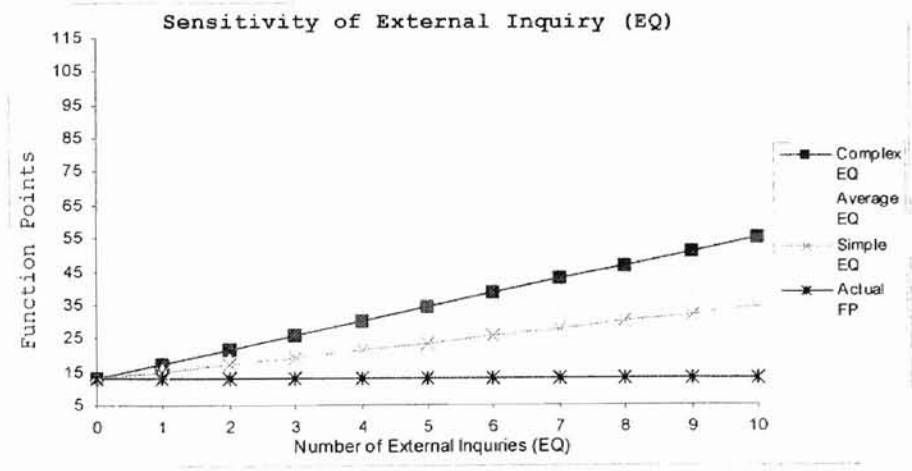
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE PAR PROGRAM

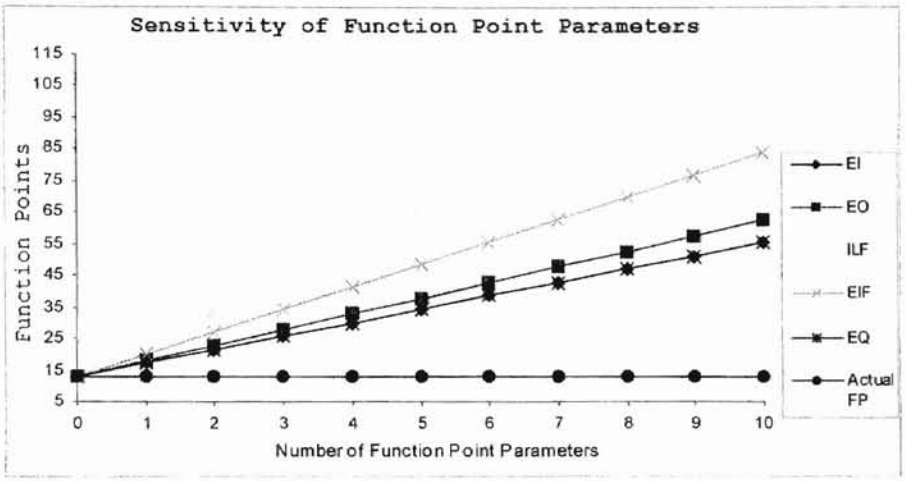


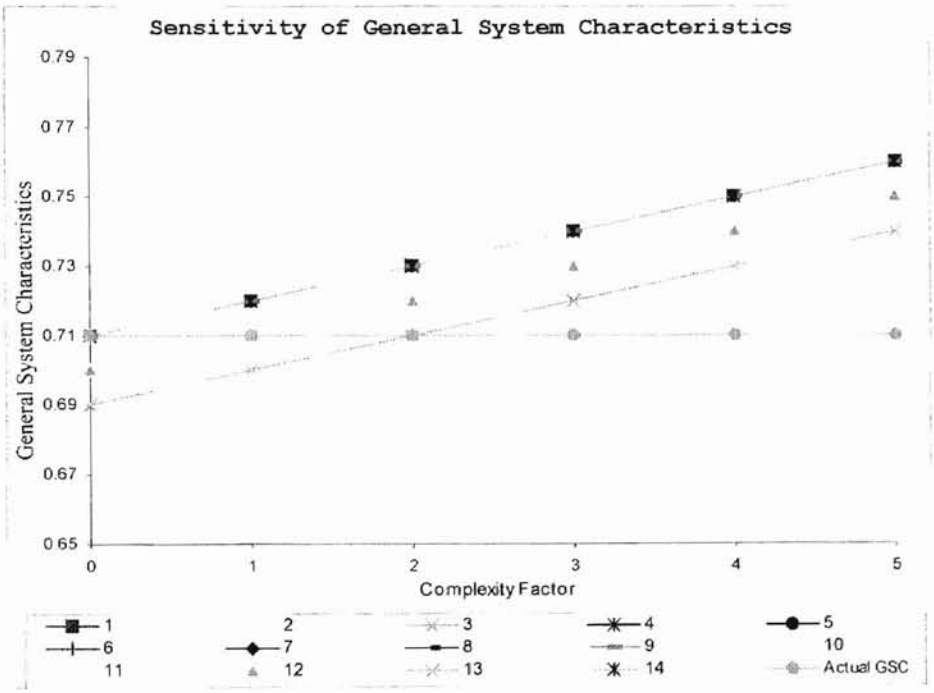



Sensitivity of Internal Logical File (ILF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

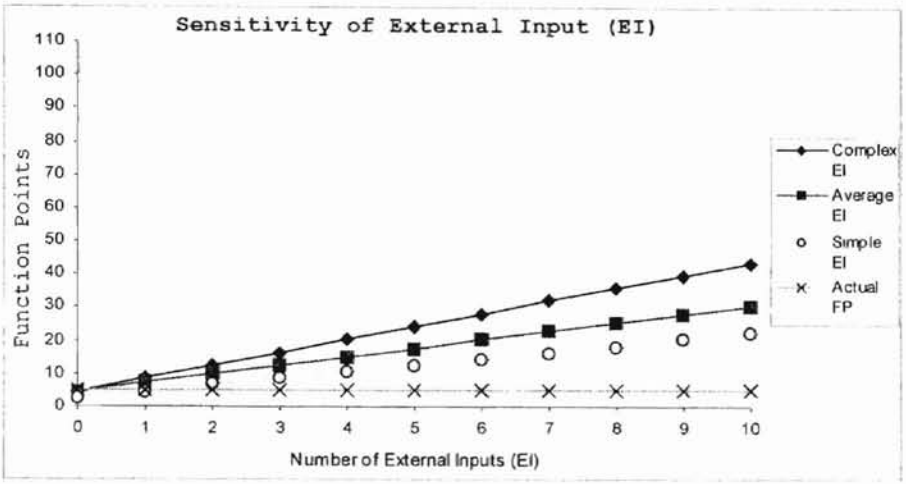

Sensitivity of External Interface File (EIF)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

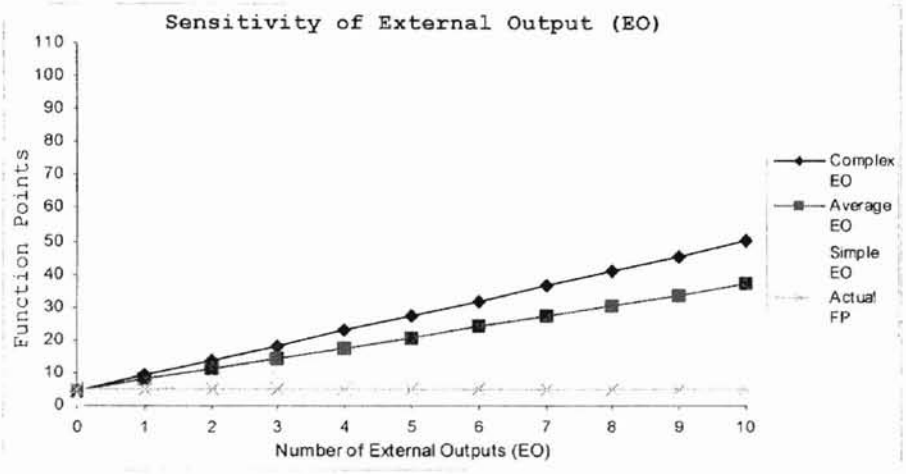

Sensitivity of External Inquiry (EQ)
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

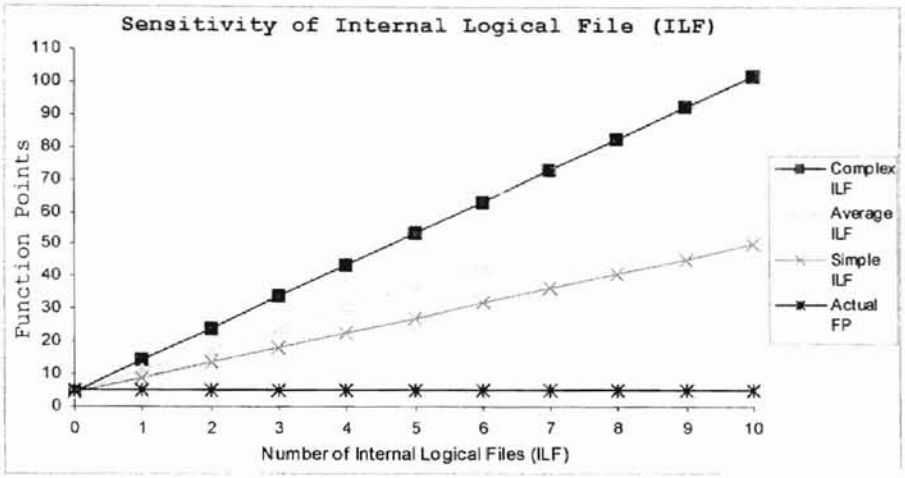

Sensitivity of Function Point Parameters
Function Points
115
105
95
85
75
65
55
45
35
25
15
5
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

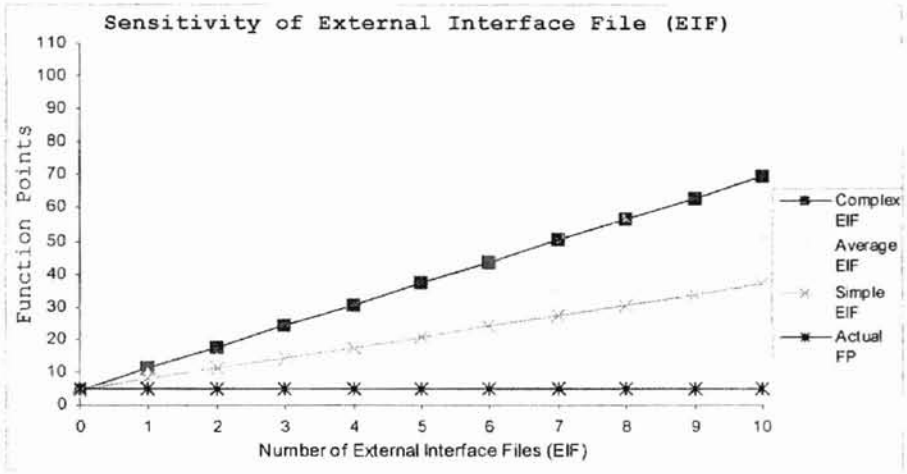

Sensitivity of General System Characteristics
General System Characteristics
0.79
0.77
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

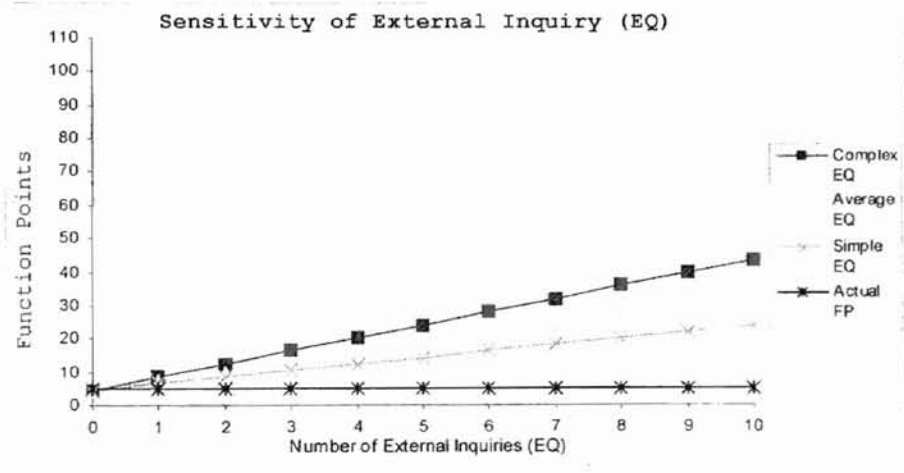
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE RS PROGRAM

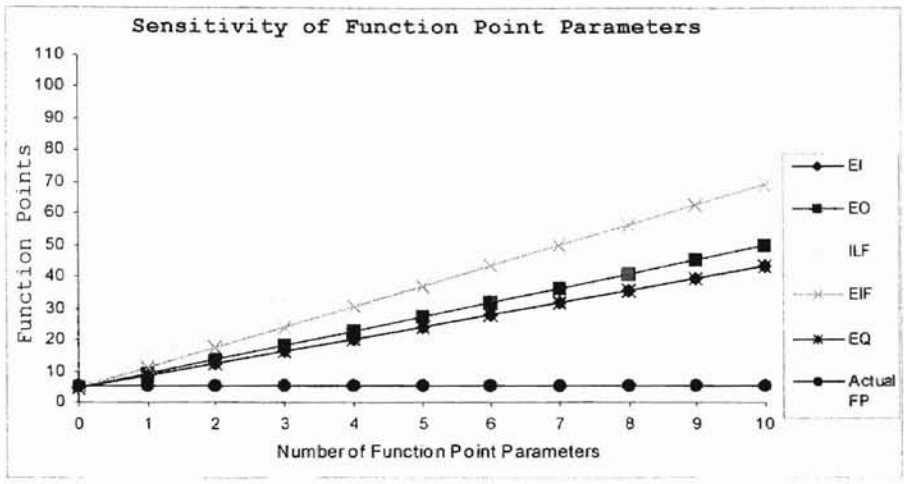


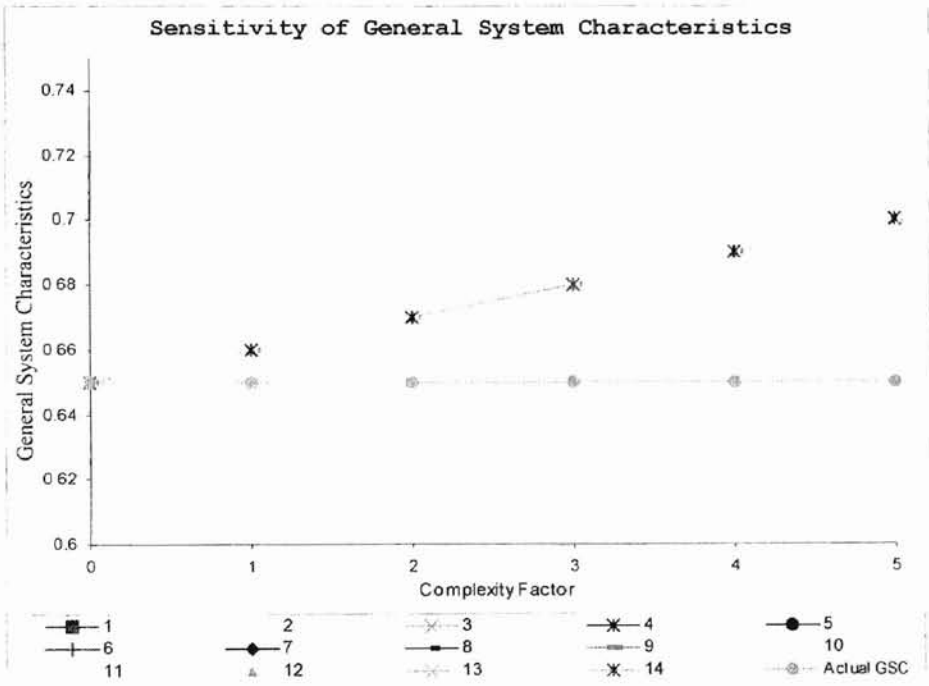



Sensitivity of Internal Logical File (ILF)
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

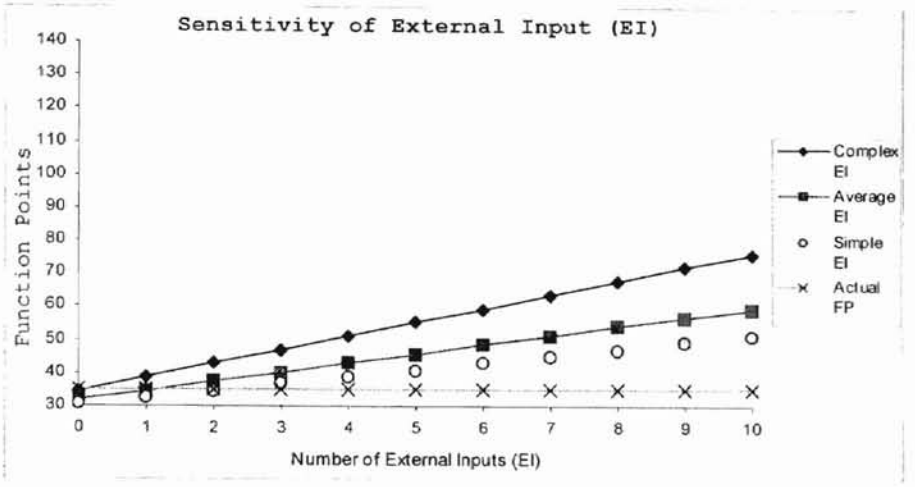

Sensitivity of External Interface File (EIF)
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

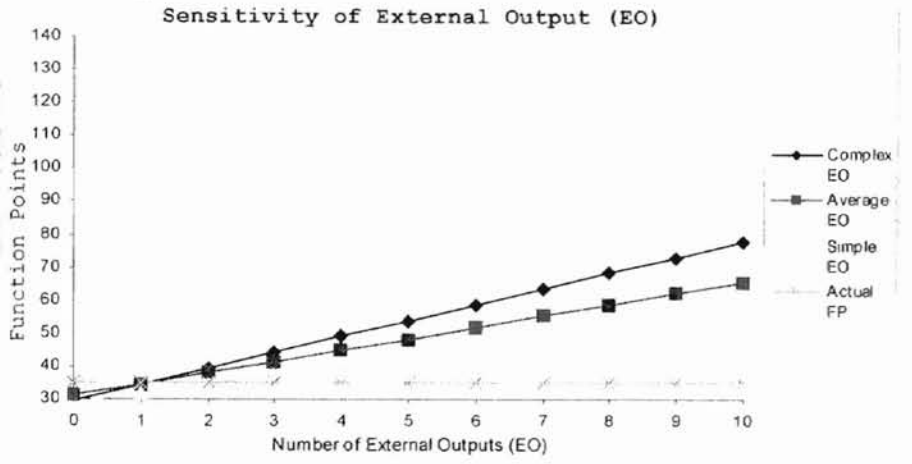

Sensitivity of External Inquiry (EQ)
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

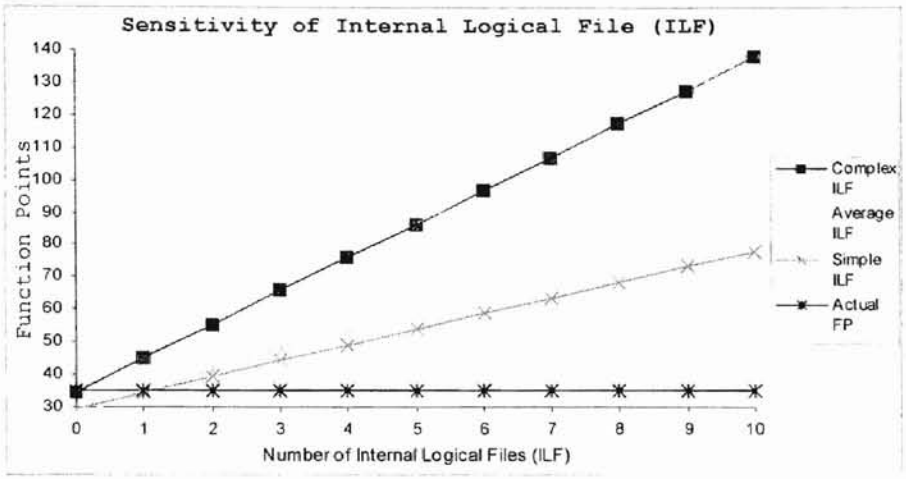

Sensitivity of Function Point Parameters
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

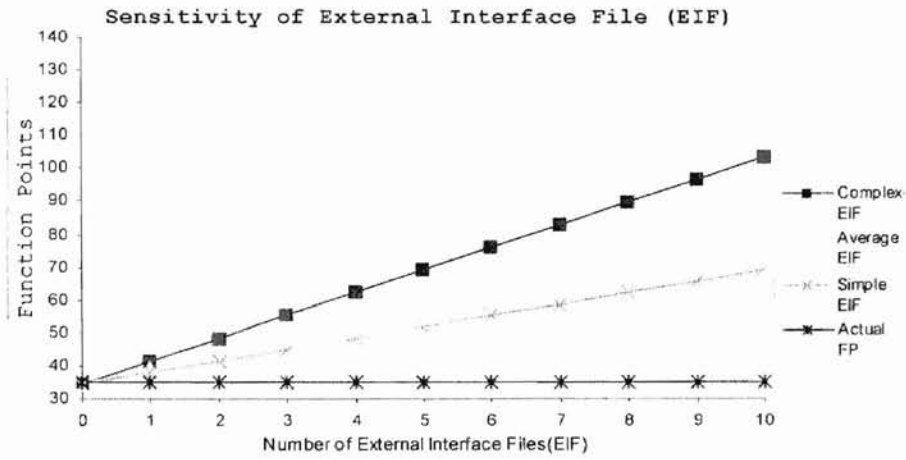

Sensitivity of General System Characteristics
General System Characteristics
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

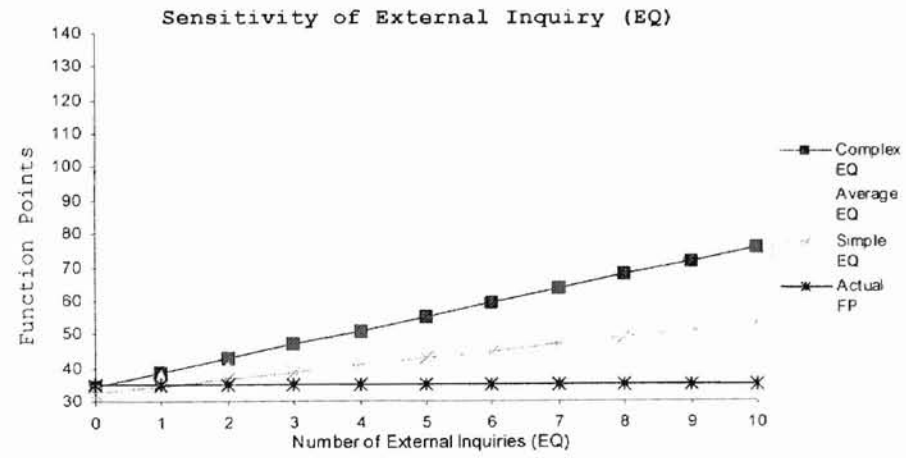
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE SAIL PROGRAM

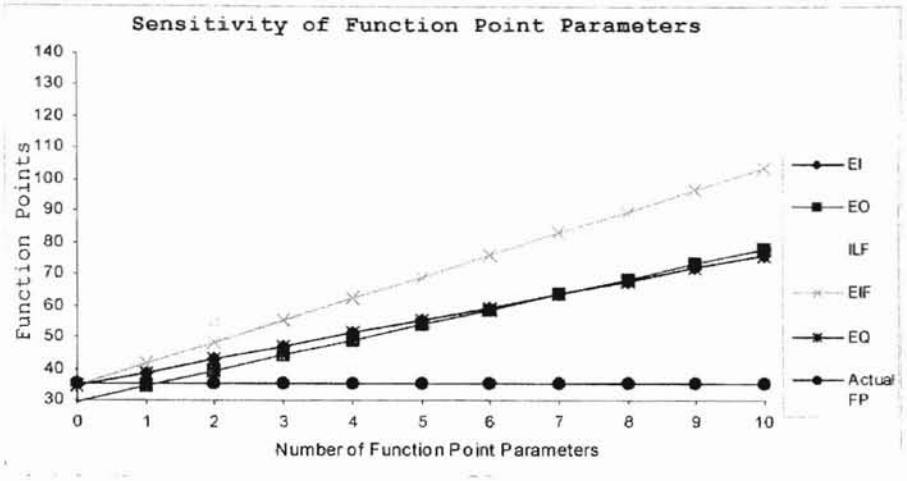


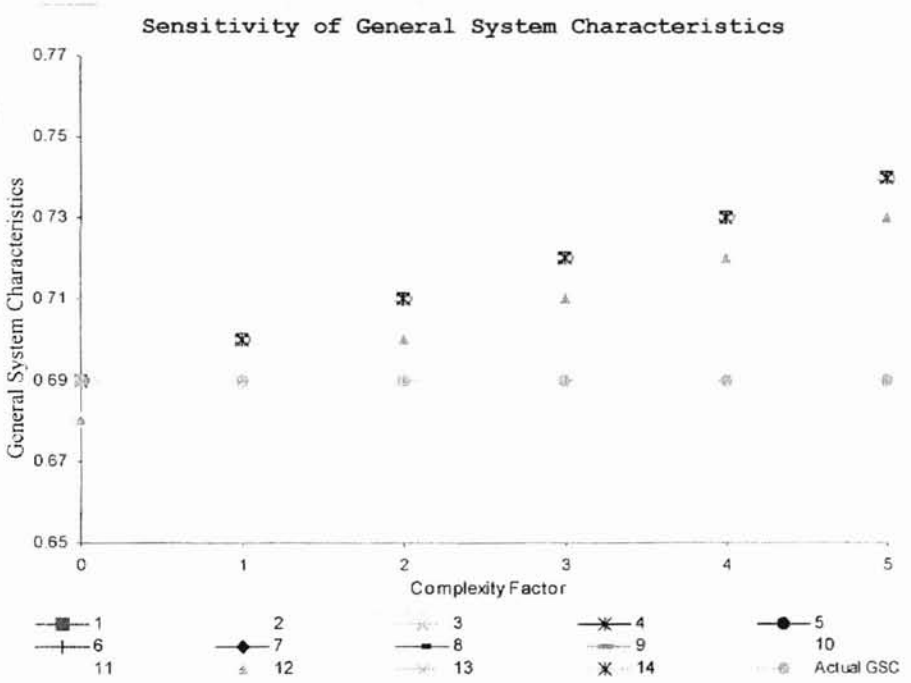



Sensitivity of Internal Logical File (ILF)
Function Points
140
130
120
110
100
90
80
70
60
50
40
30
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

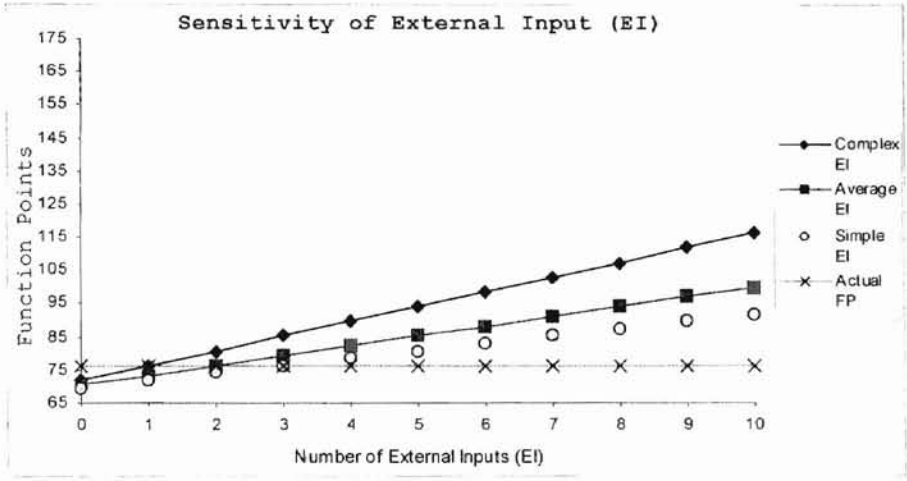

Sensitivity of External Interface File (EIF)
Function Points
140
130
120
110
100
90
80
70
60
50
40
30
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files(EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

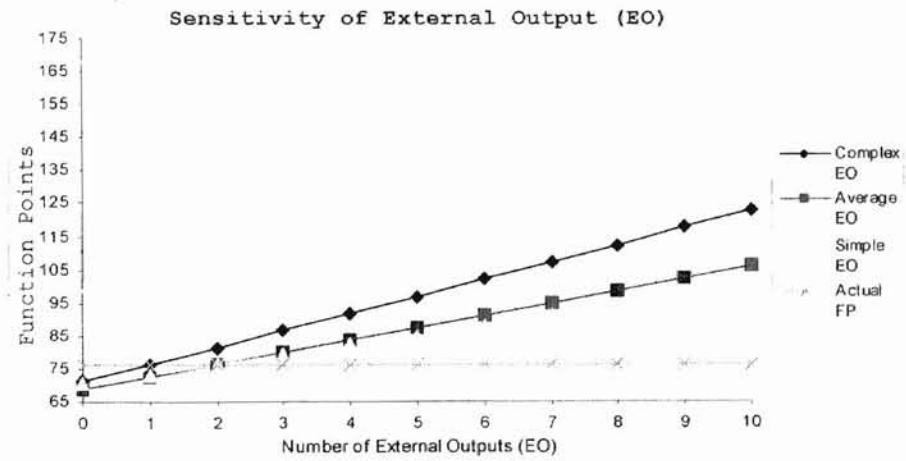

Sensitivity of External Inquiry (EQ)
Function Points
140
130
120
110
100
90
80
70
60
50
40
30
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

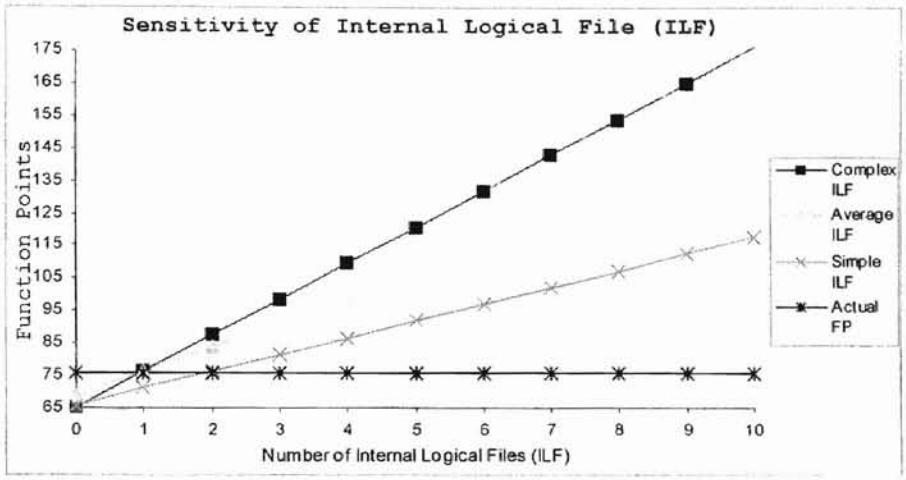

Sensitivity of Function Point Parameters
Function Points
140
130
120
110
100
90
80
70
60
50
40
30
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

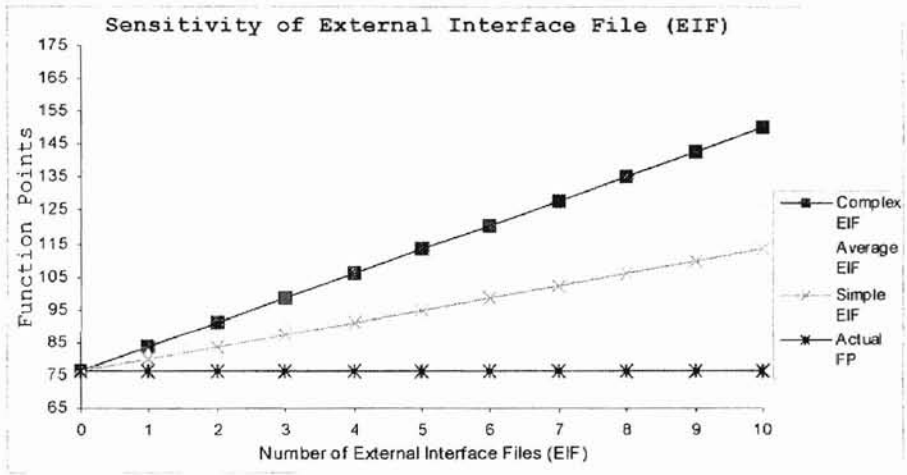

Sensitivity of General System Characteristics
General System Characteristics
0.77
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

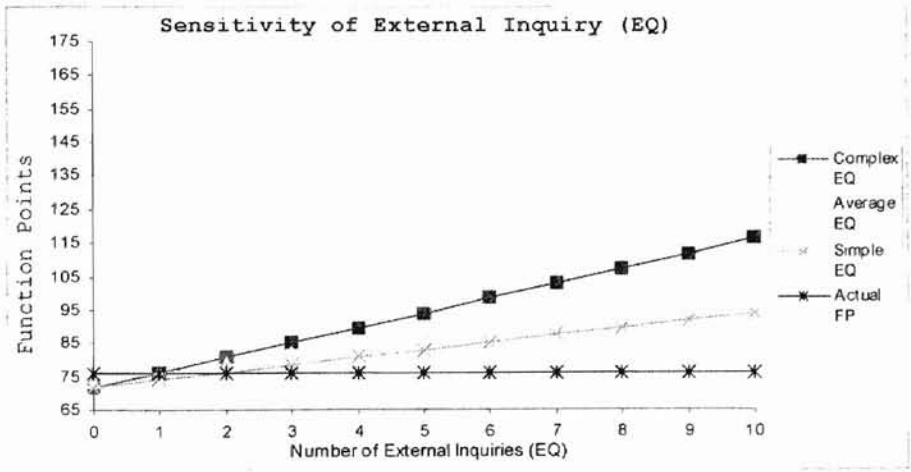
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE SC PROGRAM

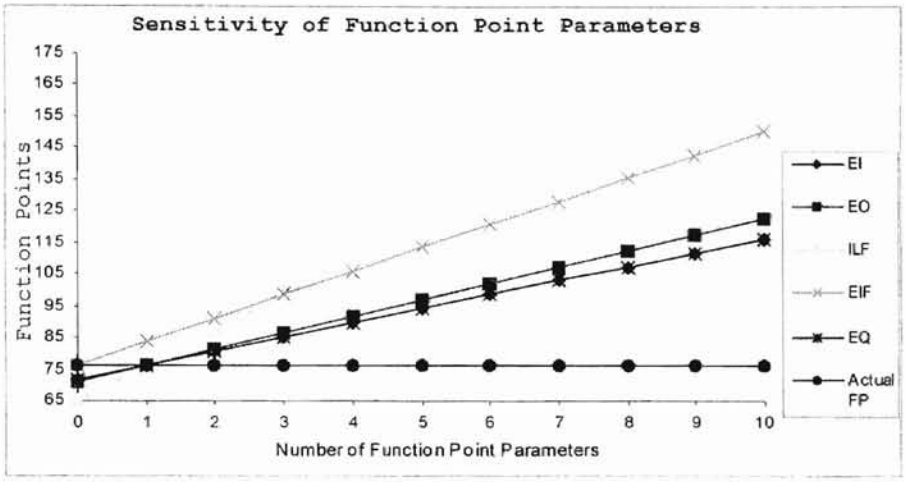


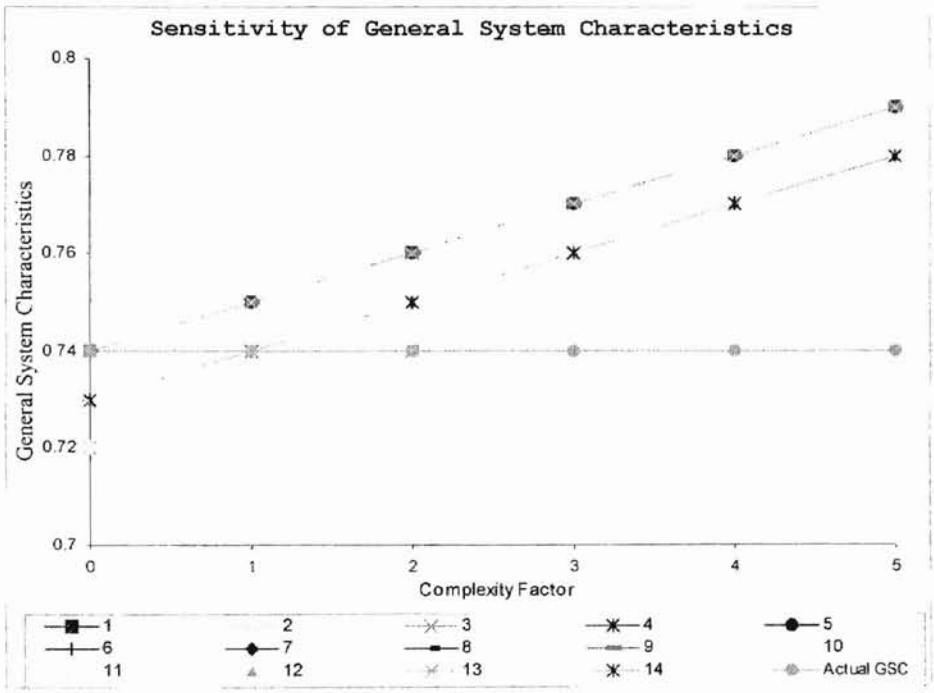



Sensitivity of Internal Logical File (ILF)
Function Points
175
165
155
145
135
125
115
105
95
85
75
65
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

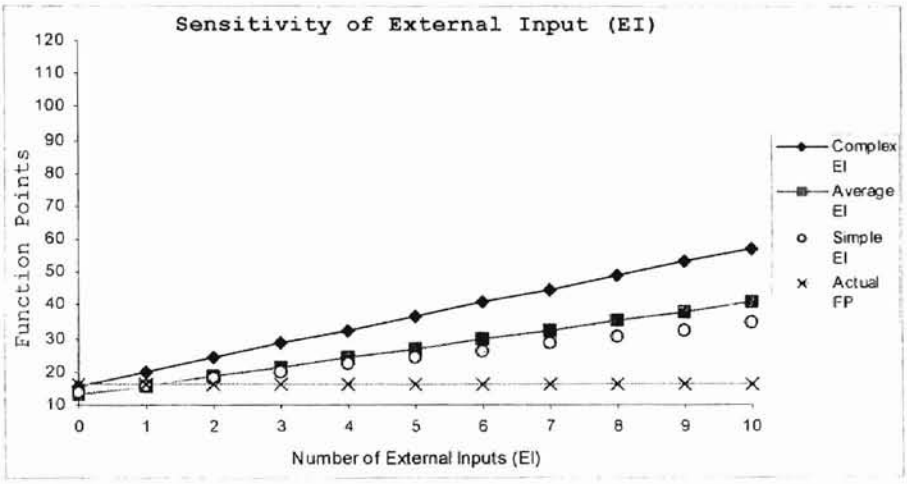

Sensitivity of External Interface File (EIF)
Function Points
175
165
155
145
135
125
115
105
95
85
75
65
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

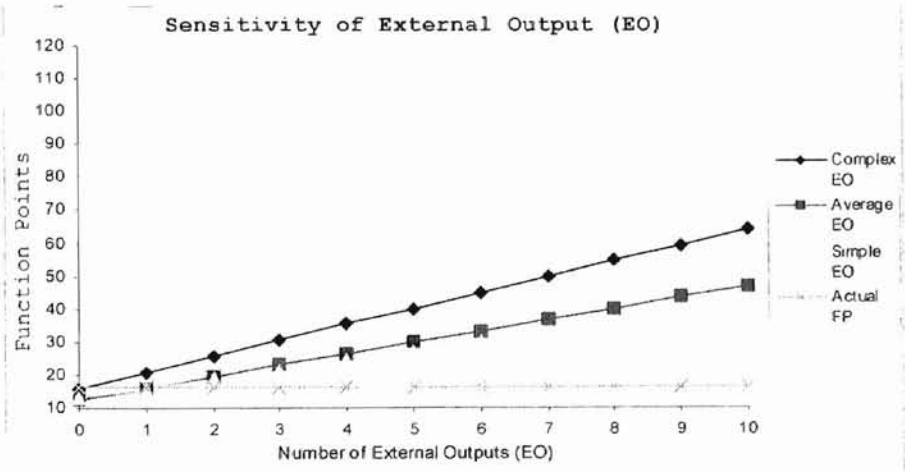

Sensitivity of External Inquiry (EQ)
Function Points
175
165
155
145
135
125
115
105
95
85
75
65
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

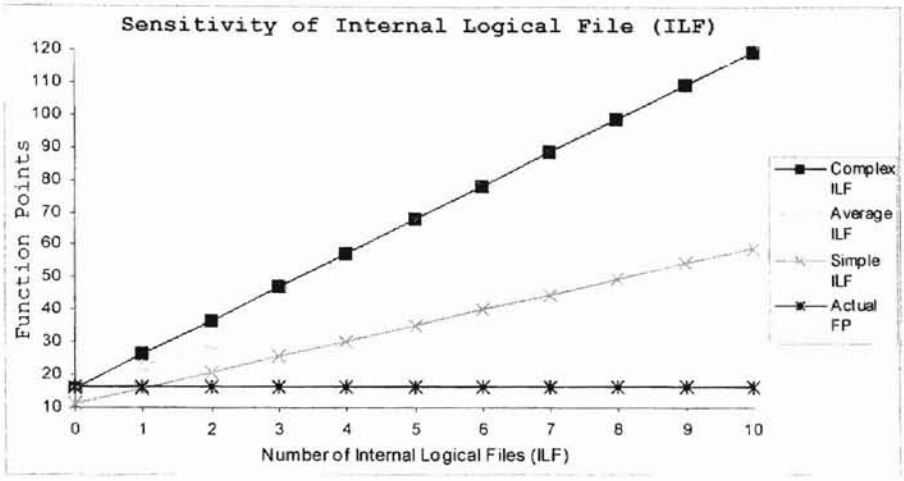

Sensitivity of Function Point Parameters
Function Points
175
165
155
145
135
125
115
105
95
85
75
65
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

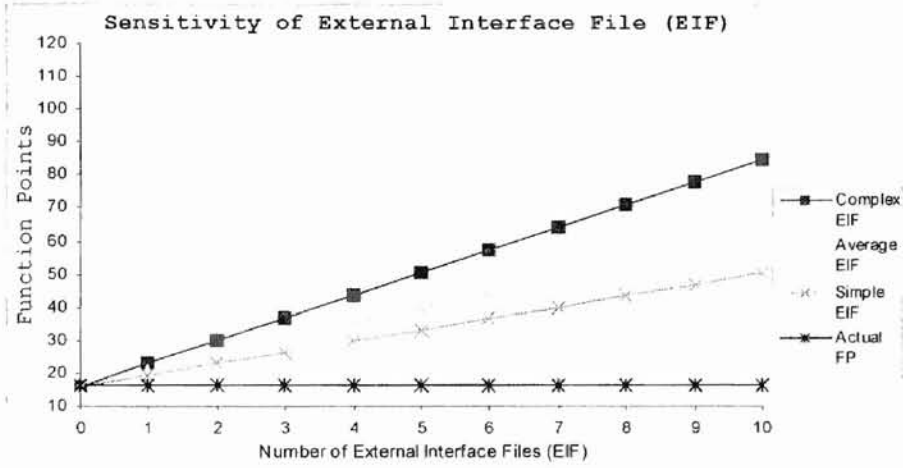

Sensitivity of General System Characteristics
General System Characteristics
0.8
0.78
0.76
0.74
0.72
0.7
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

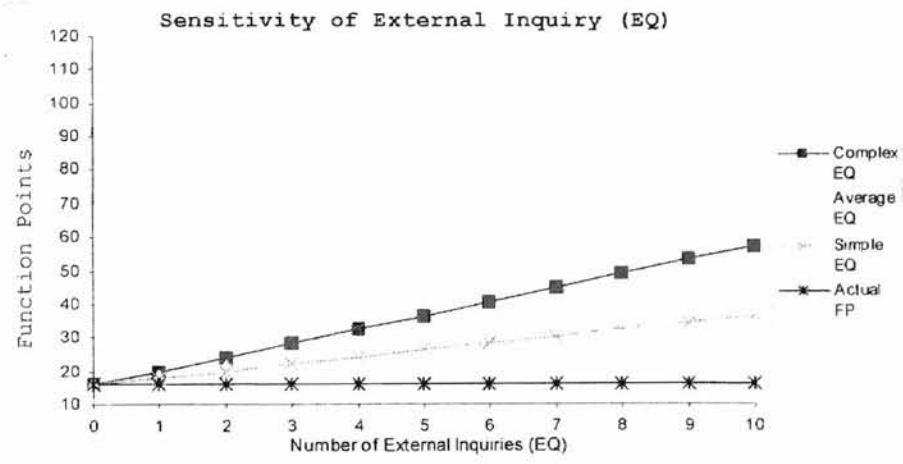
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE SEGMENT PROGRAM

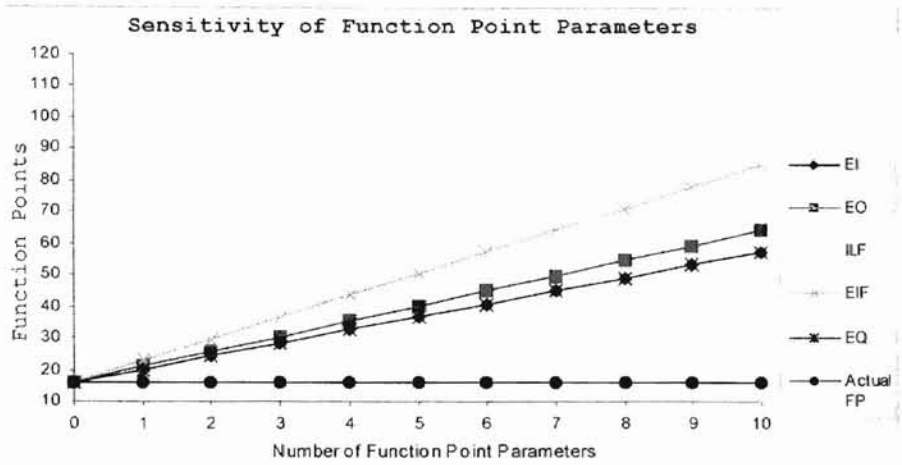


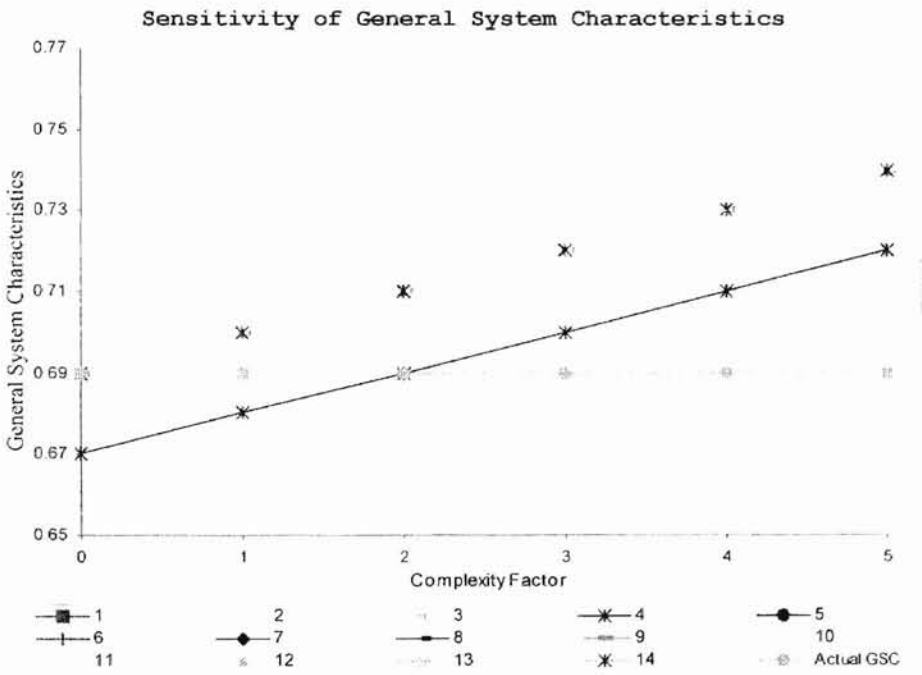



Sensitivity of Internal Logical File (ILF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

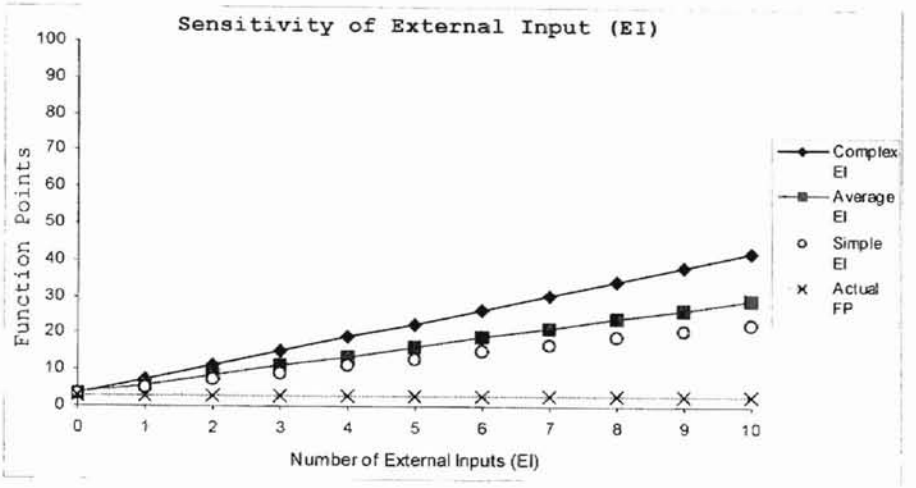

Sensitivity of External Interface File (EIF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

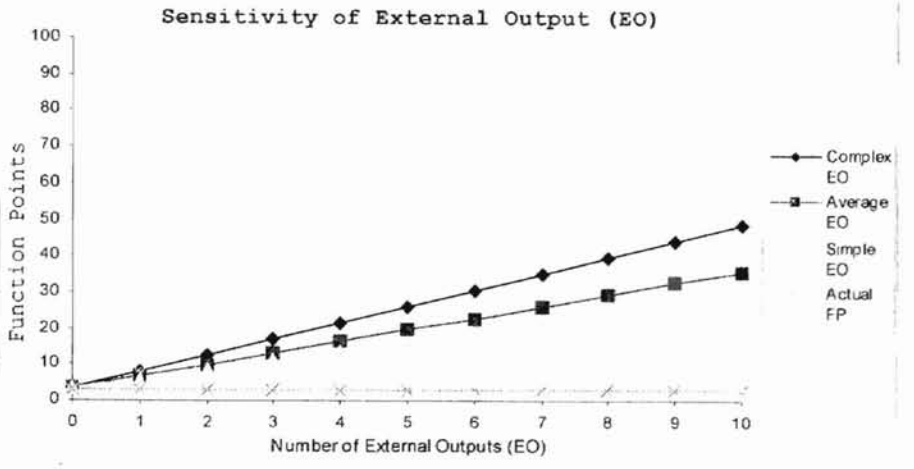

Sensitivity of External Inquiry (EQ)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

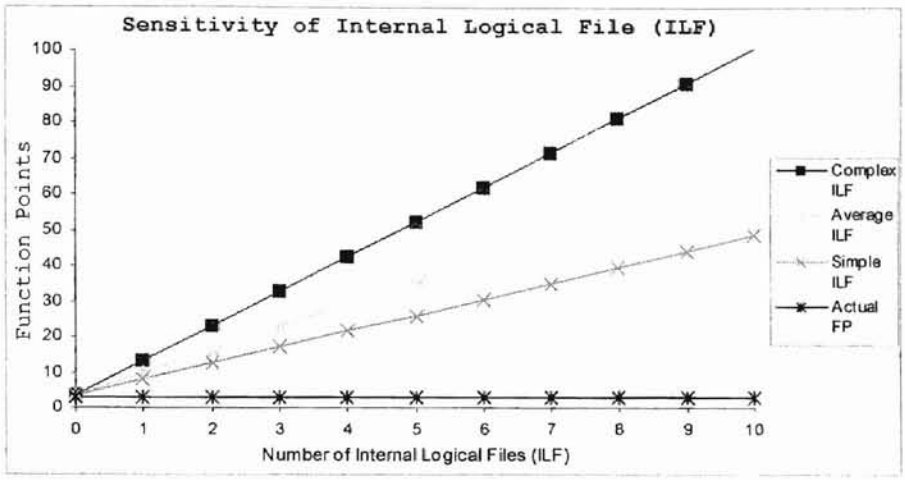

Sensitivity of Function Point Parameters
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

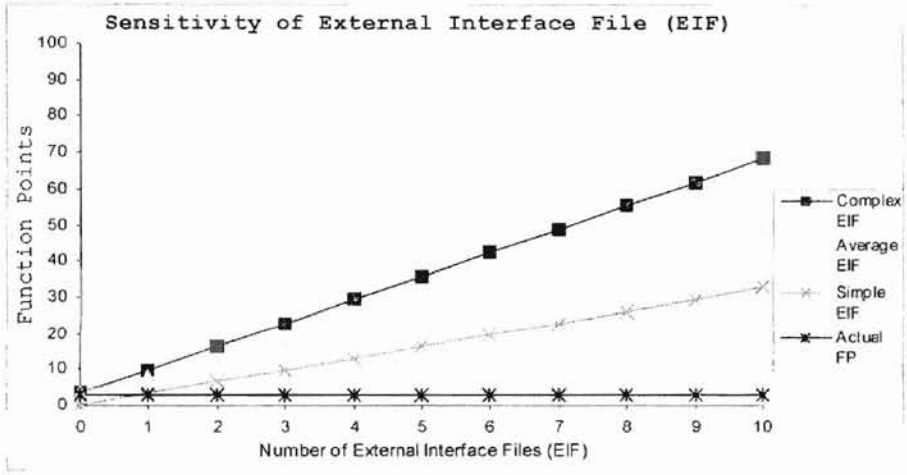

Sensitivity of General System Characteristics
General System Characteristics
0.77
0.75
0.73
0.71
0.69
0.67
0.65
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

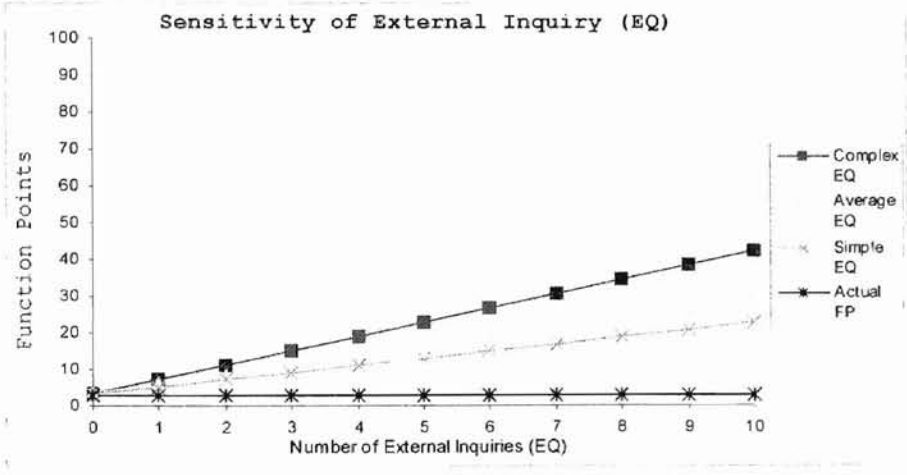
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE STAT PROGRAM

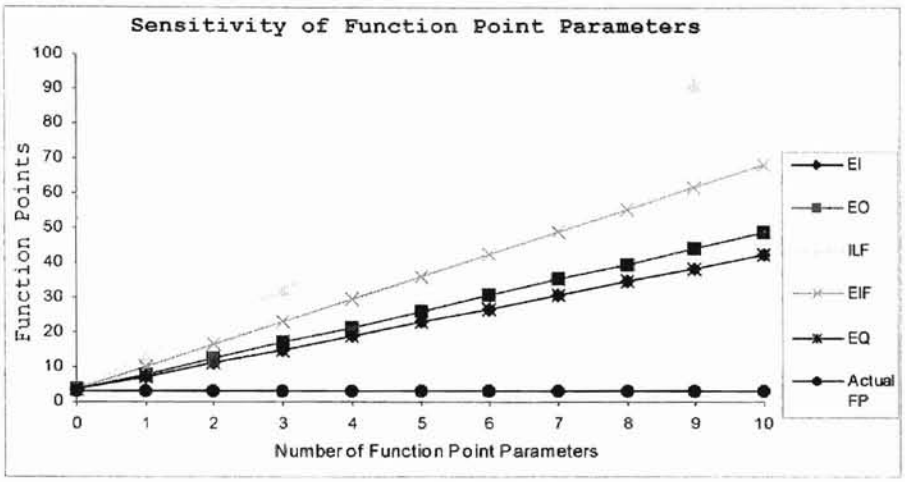


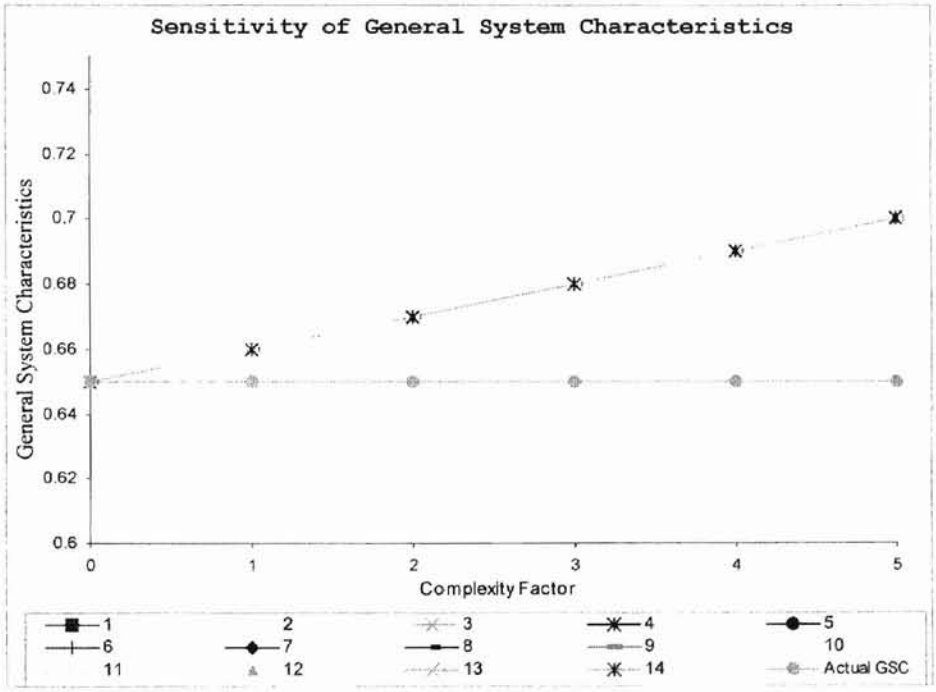



Sensitivity of Internal Logical File (ILF)
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

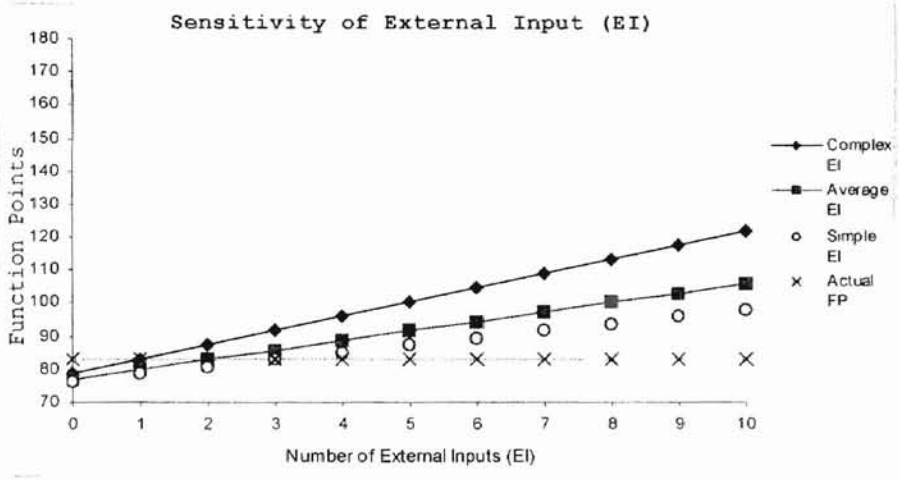

Sensitivity of External Interface File (EIF)
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

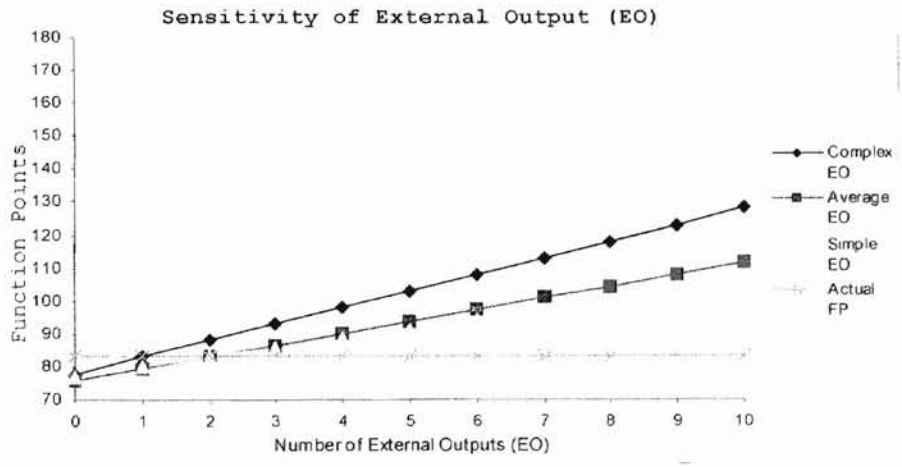

Sensitivity of External Inquiry (EQ)
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

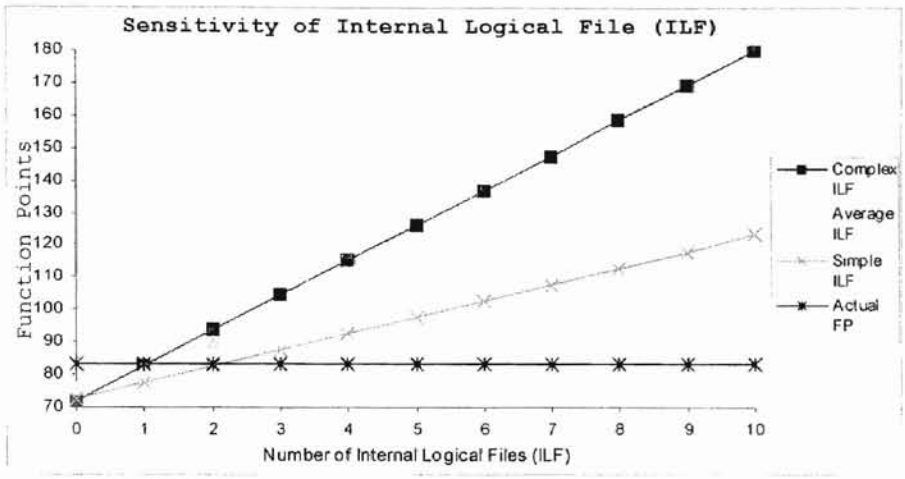

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

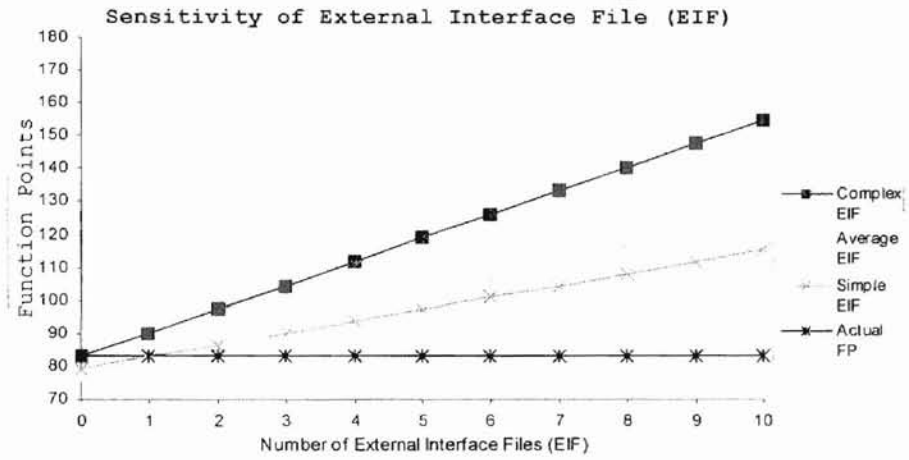

Sensitivity of General System Characteristics
General System Characteristics
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

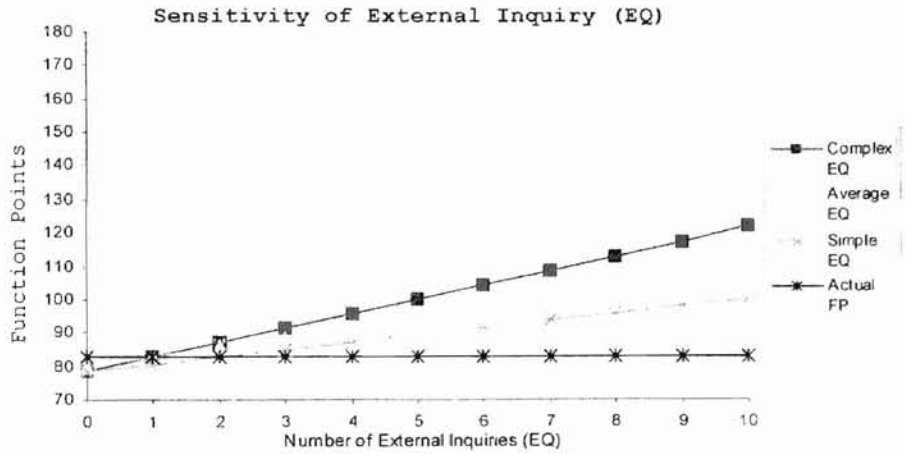
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE UTREE PROGRAM

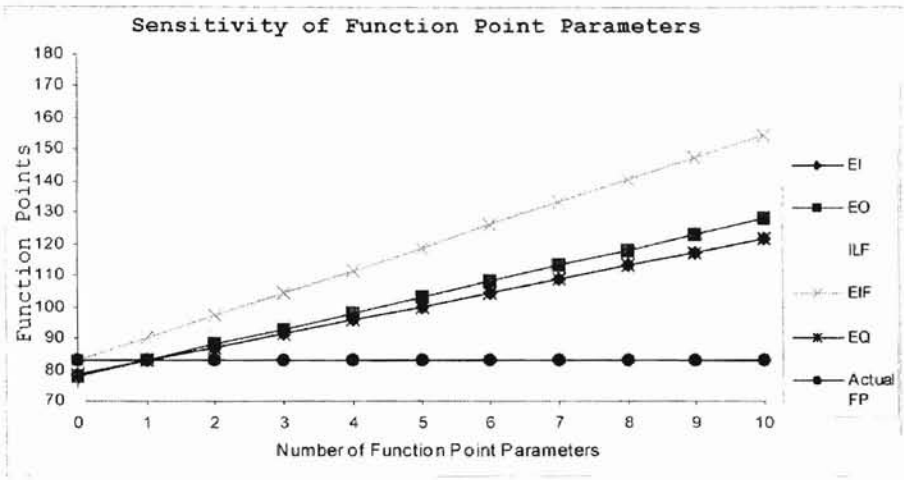


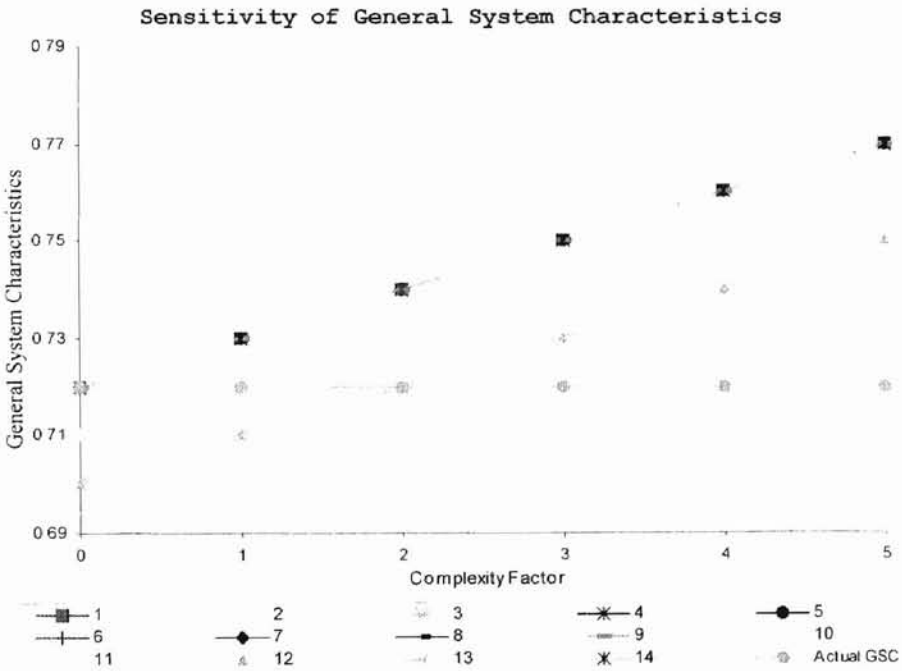



Sensitivity of Internal Logical File (ILF)
Function Points
180
170
160
150
140
130
120
110
100
90
80
70
0
1
2
3
4
5
6
7
8
9
10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

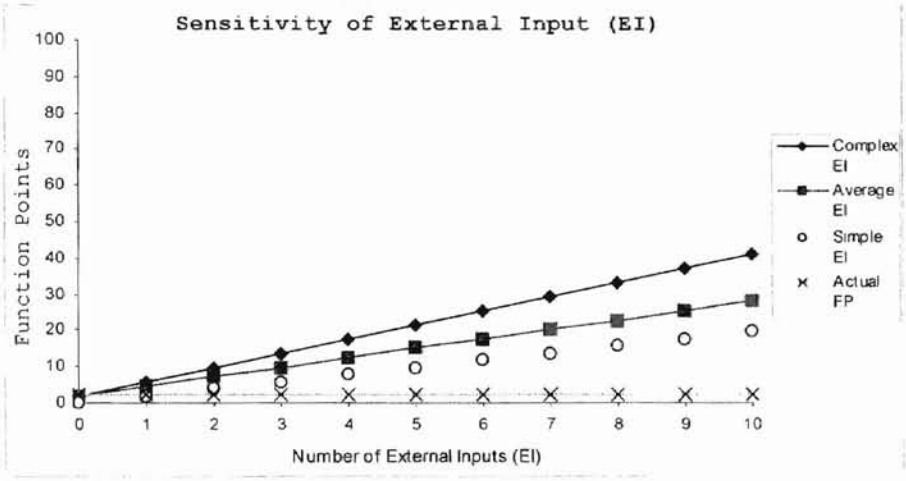

Sensitivity of External Interface File (EIF)
Function Points
180
170
160
150
140
130
120
110
100
90
80
70
0
1
2
3
4
5
6
7
8
9
10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

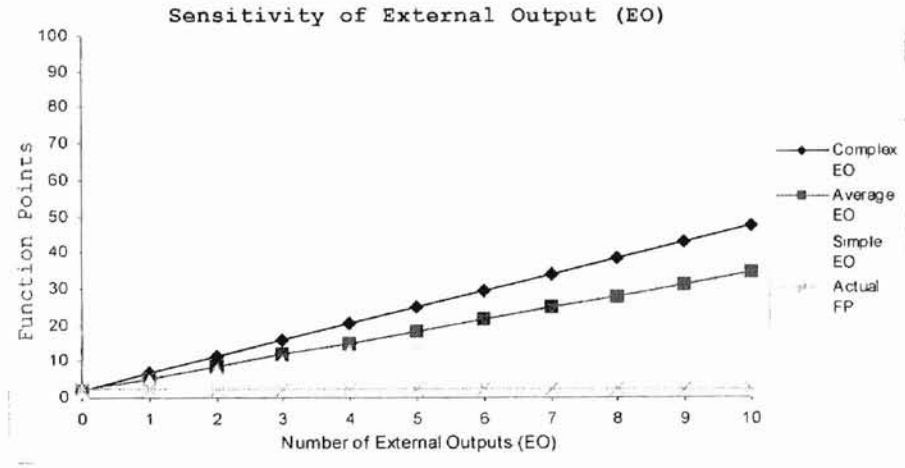

Sensitivity of External Inquiry (EQ)
Function Points
180
170
160
150
140
130
120
110
100
90
80
70
0
1
2
3
4
5
6
7
8
9
10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

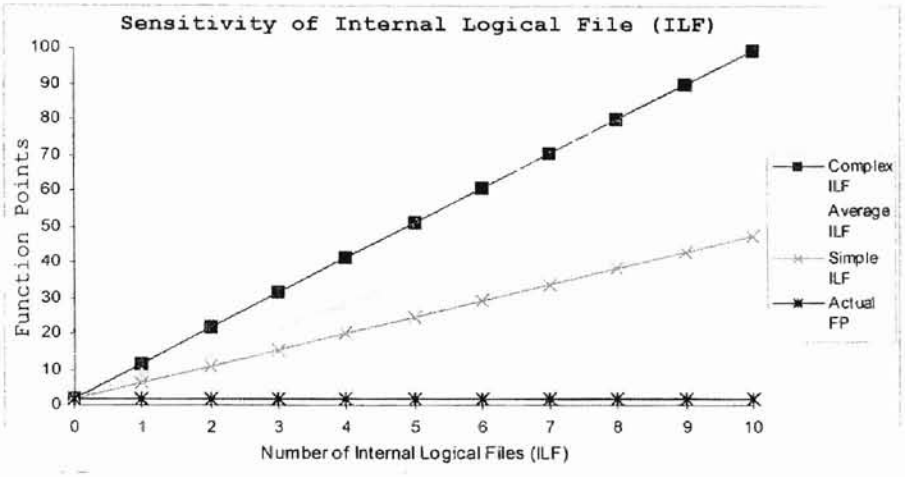

Sensitivity of Function Point Parameters
Function Points
180
170
160
150
140
130
120
110
100
90
80
70
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

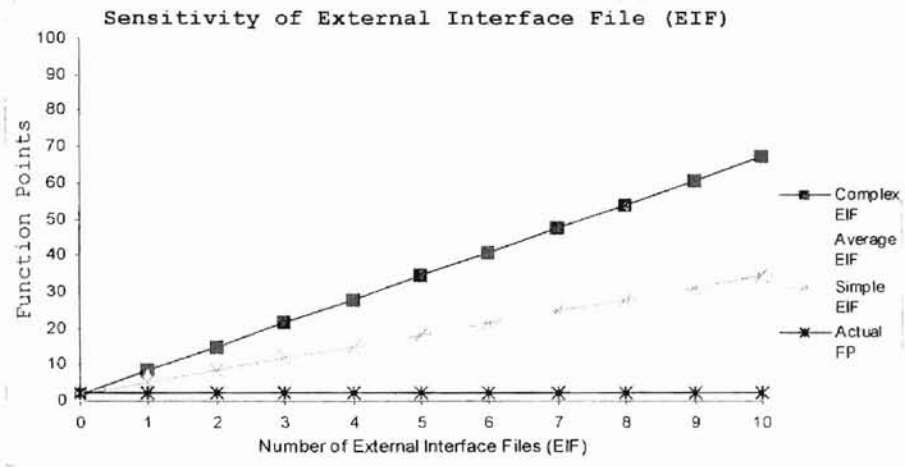

Sensitivity of General System Characteristics
General System Characteristics
0 79
0 77
0 75
0 73
0 71
0 69
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

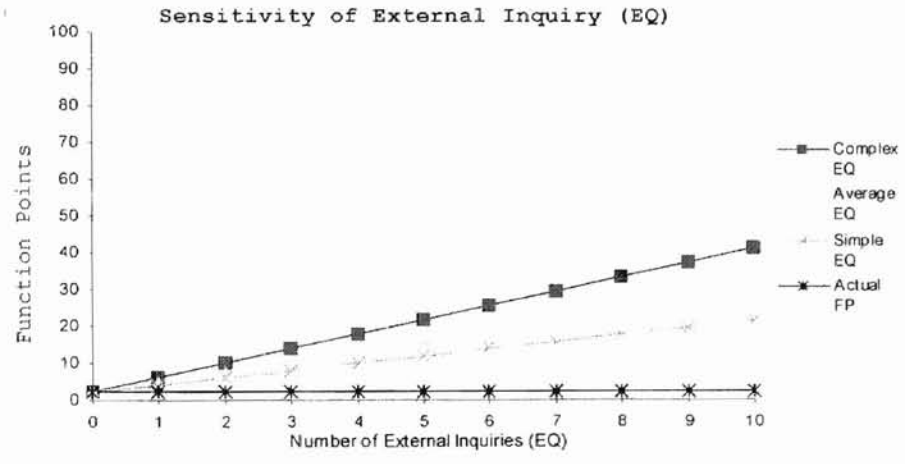
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE UUCONVERT PROGRAM

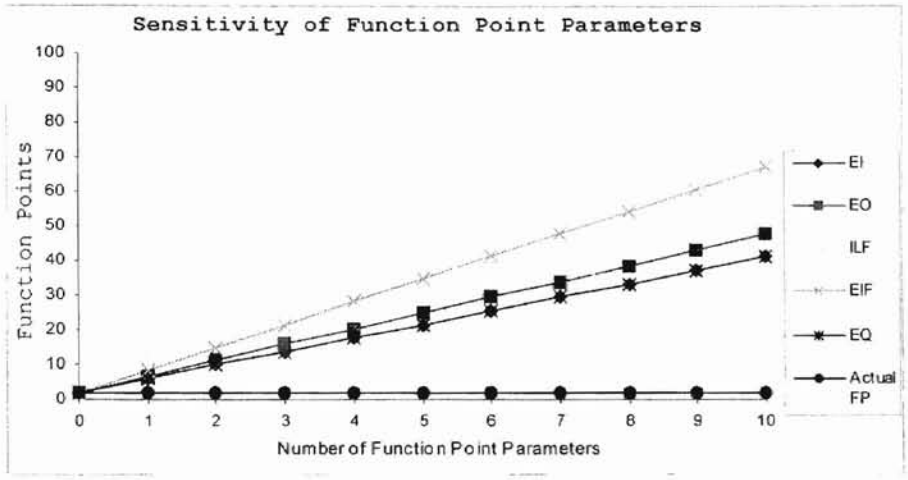


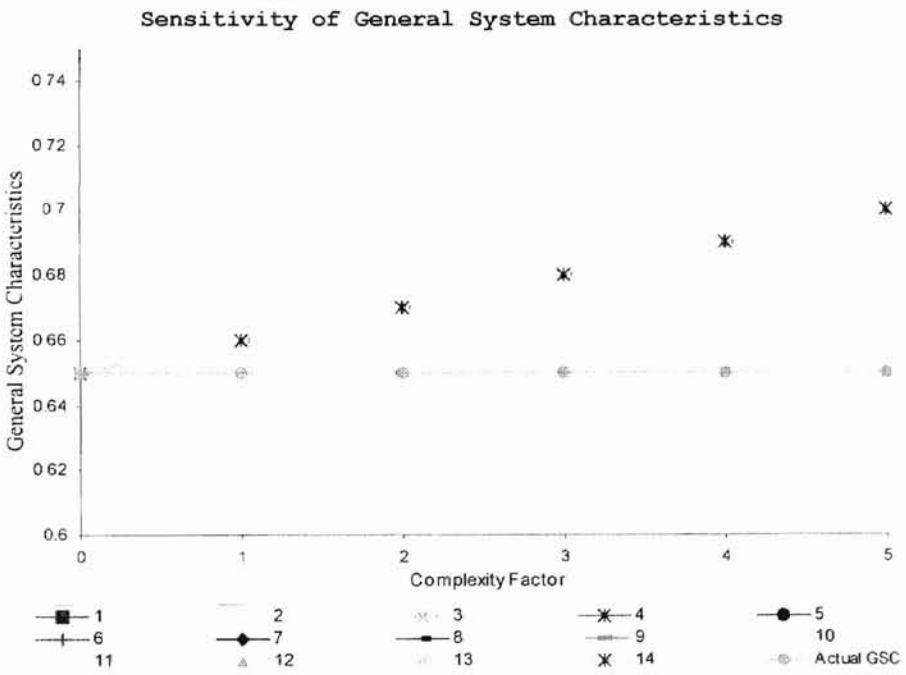



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

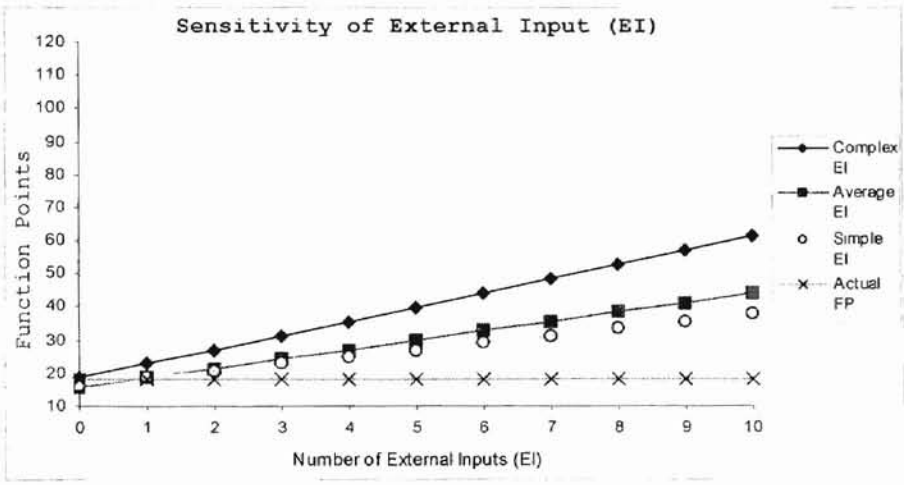

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

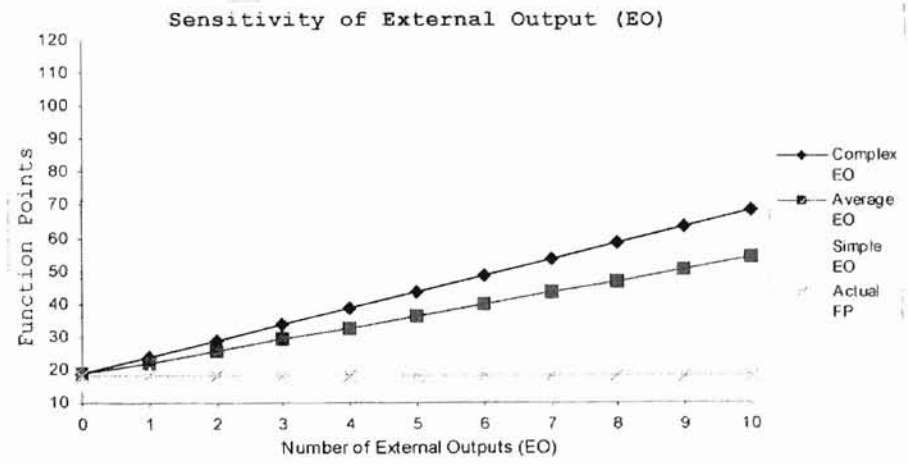

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

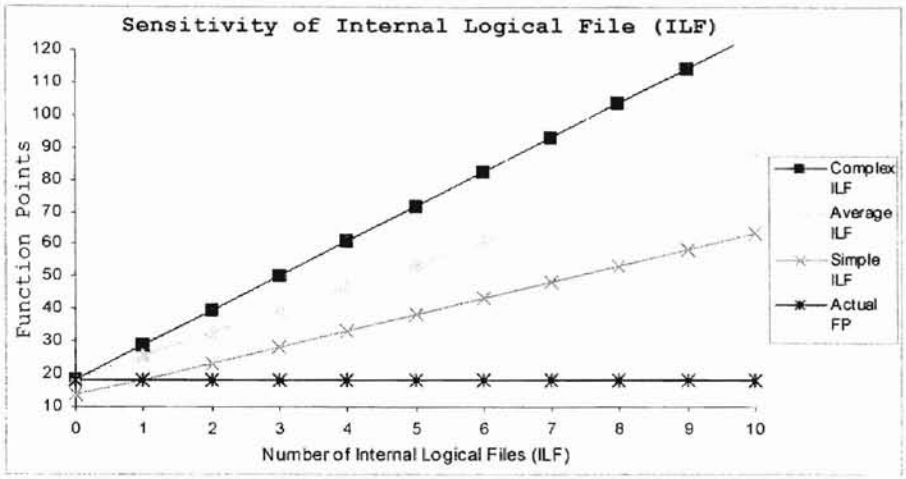

Sensitivity of Function Point Parameters
Function Points
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

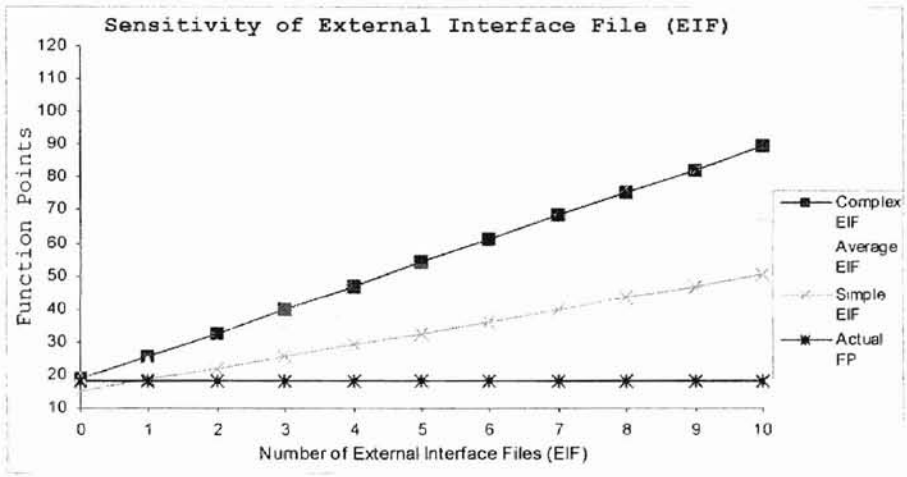

Sensitivity of General System Characteristics
General System Characteristics
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

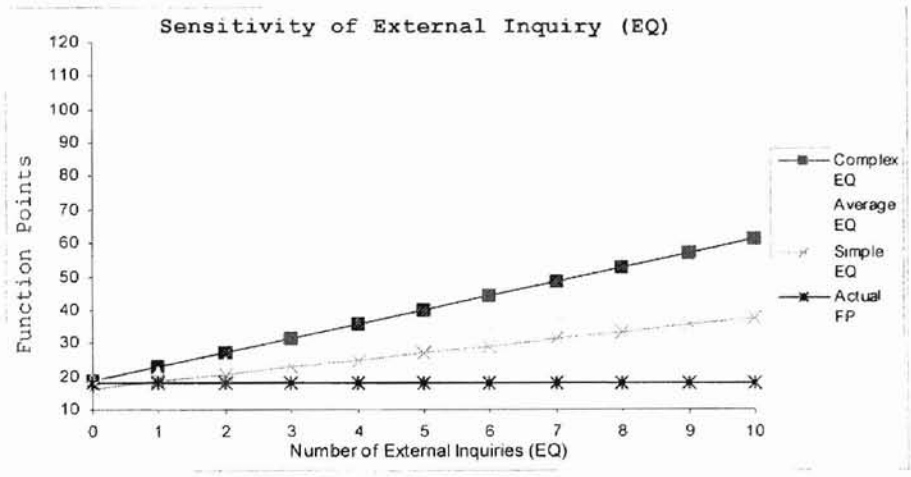
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XALARM PROGRAM

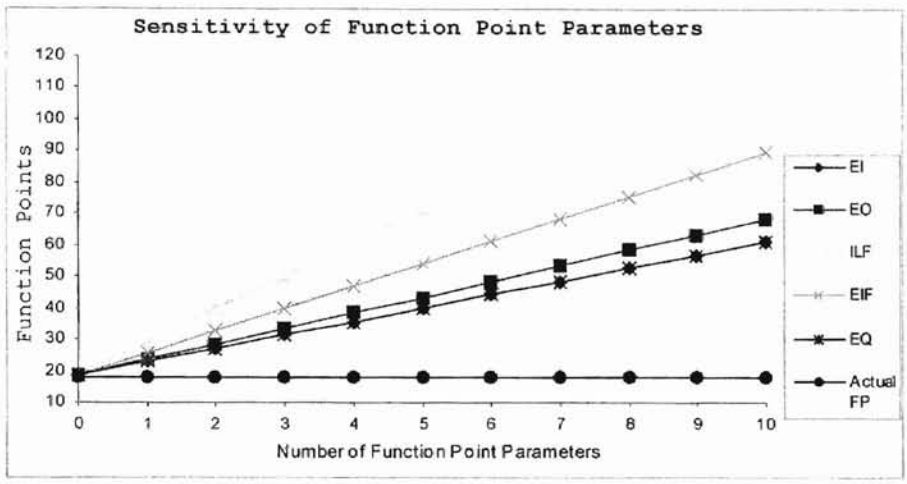


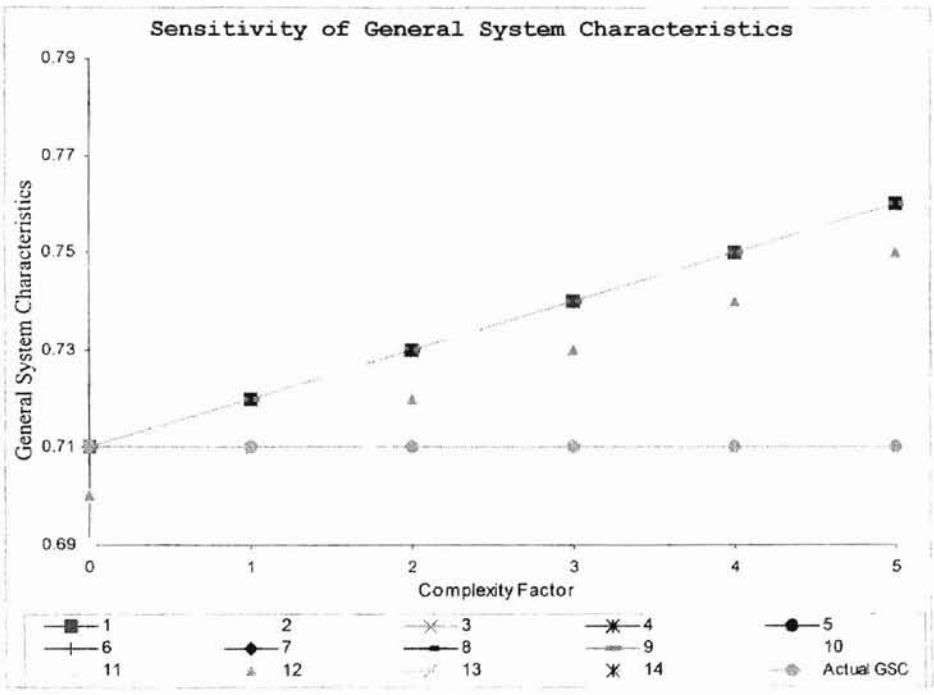



Sensitivity of Internal Logical File (ILF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

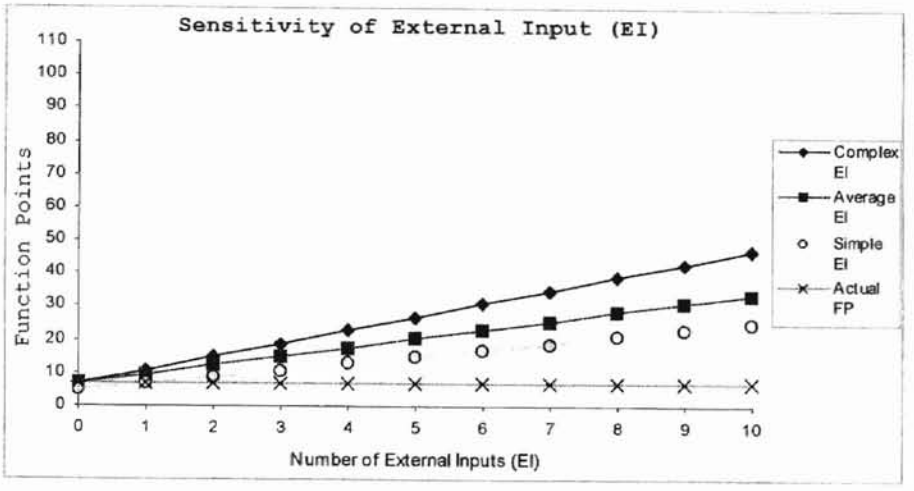

Sensitivity of External Interface File (EIF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

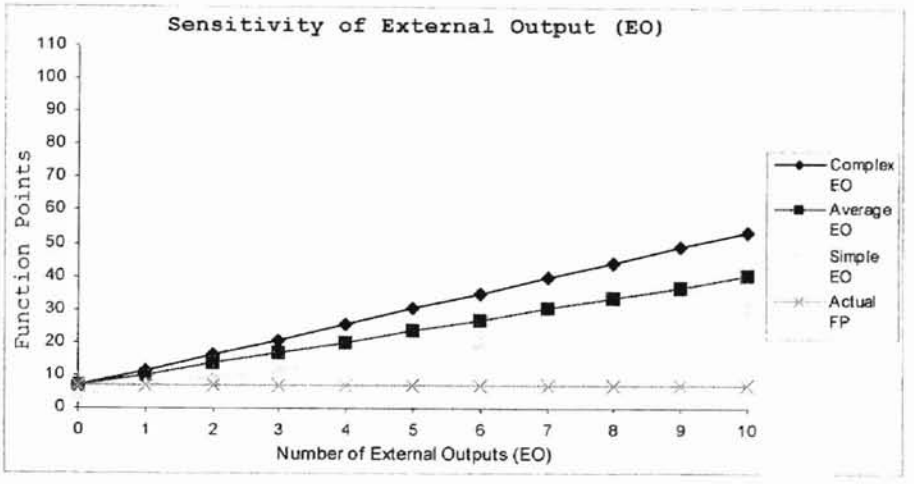

Sensitivity of External Inquiry (EQ)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

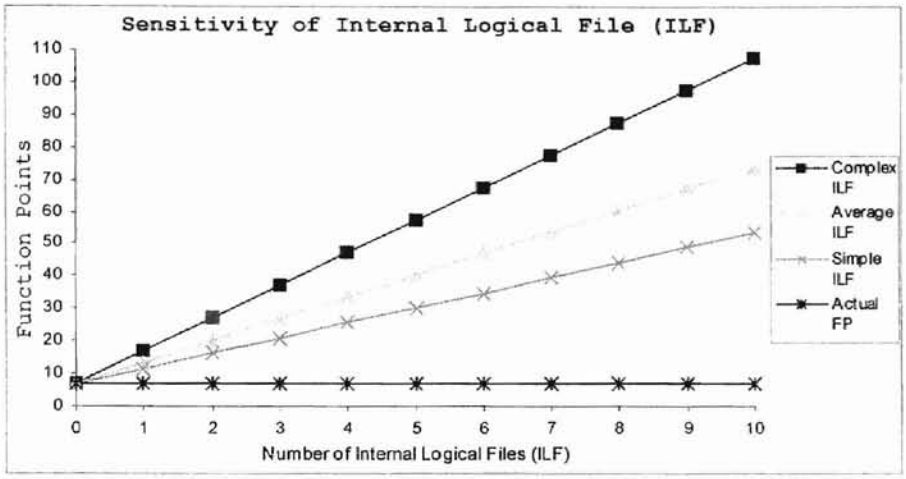

Sensitivity of Function Point Parameters
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

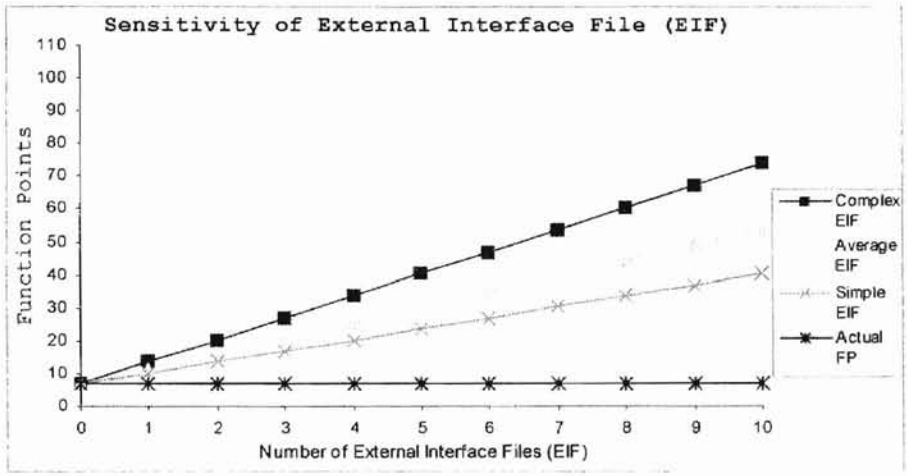

Sensitivity of General System Characteristics
General System Characteristics
0.79
0.77
0.75
0.73
0.71
0.69
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

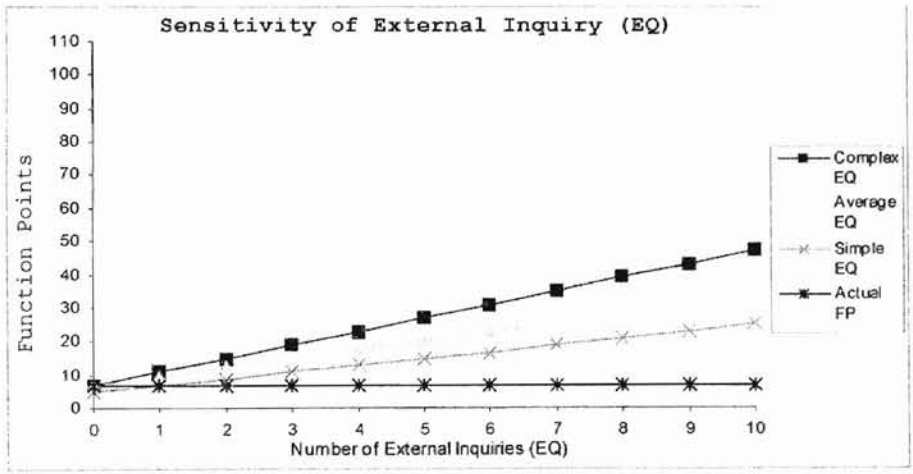
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XASTEROIDS PROGRAM

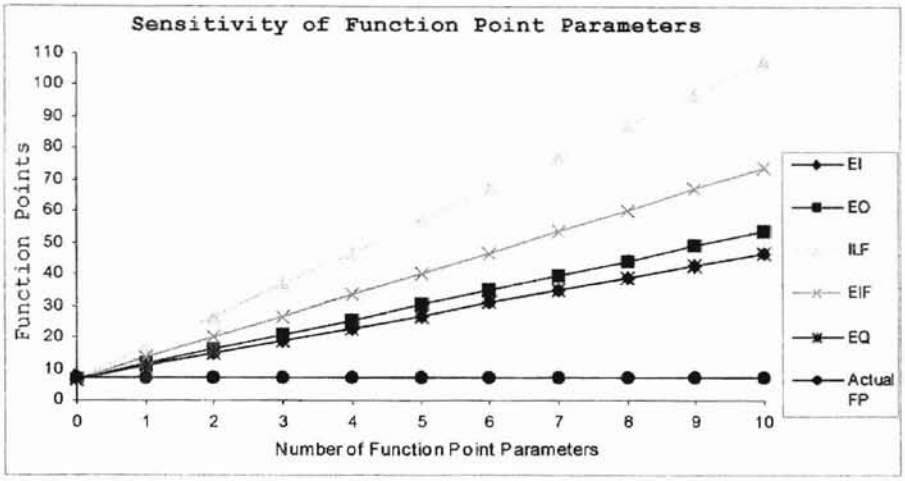


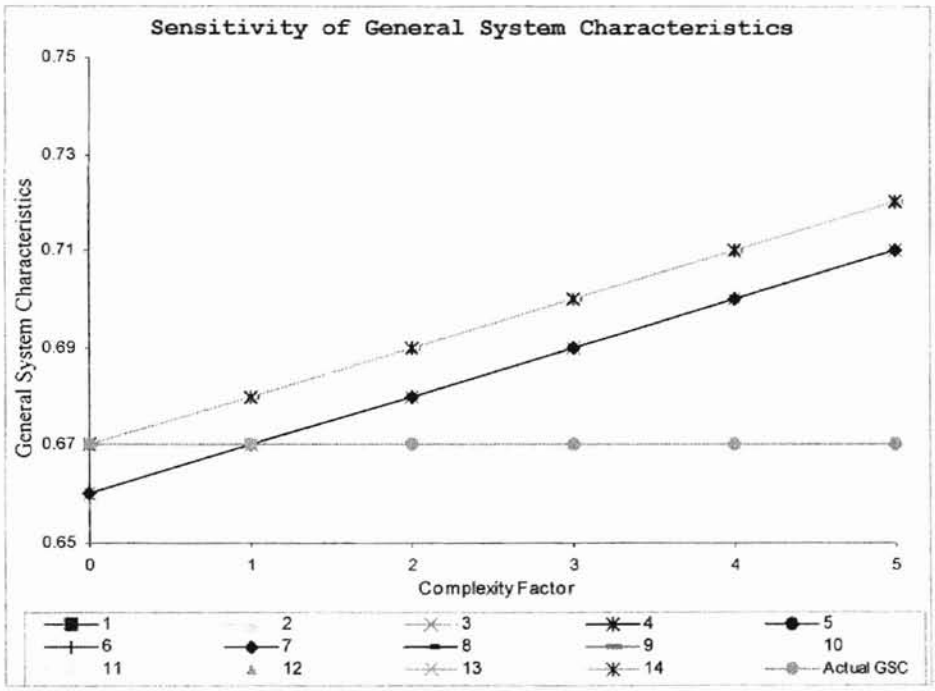



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

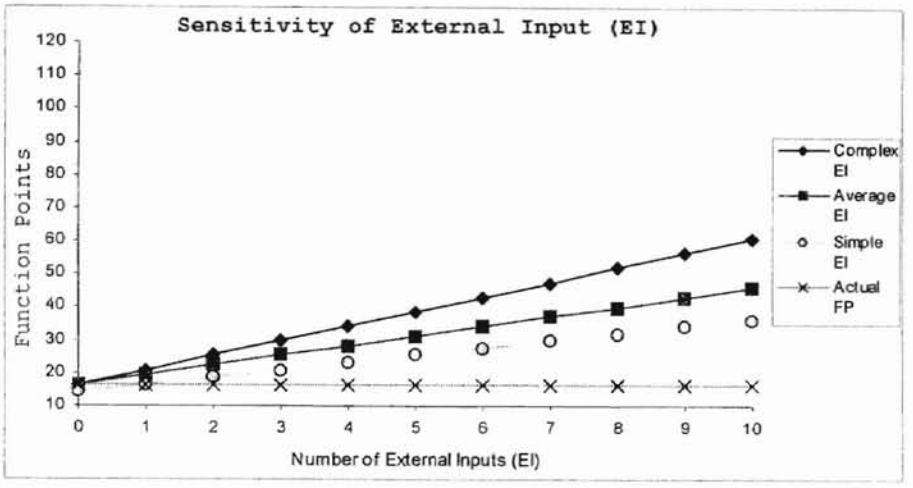

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

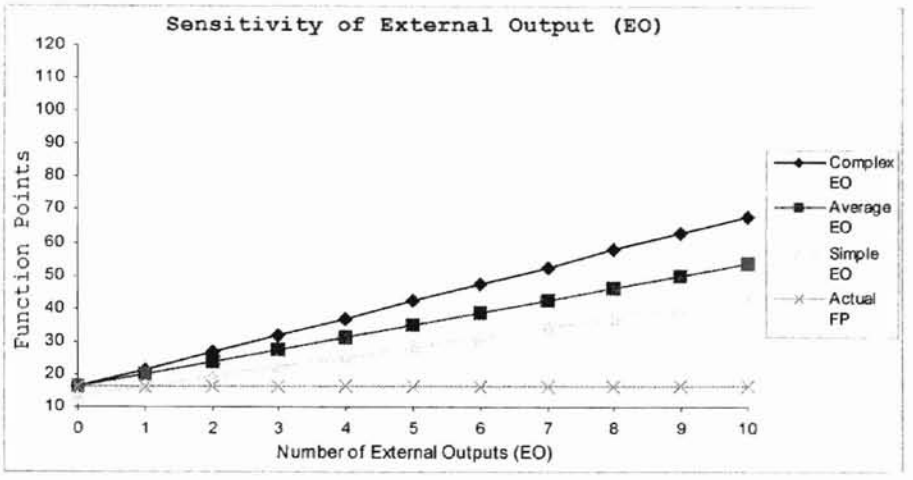

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

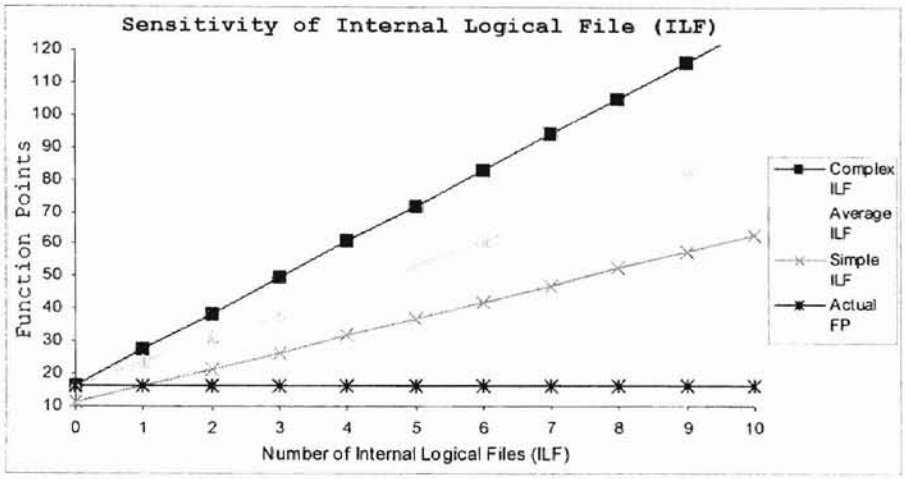

Sensitivity of Function Point Parameters
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

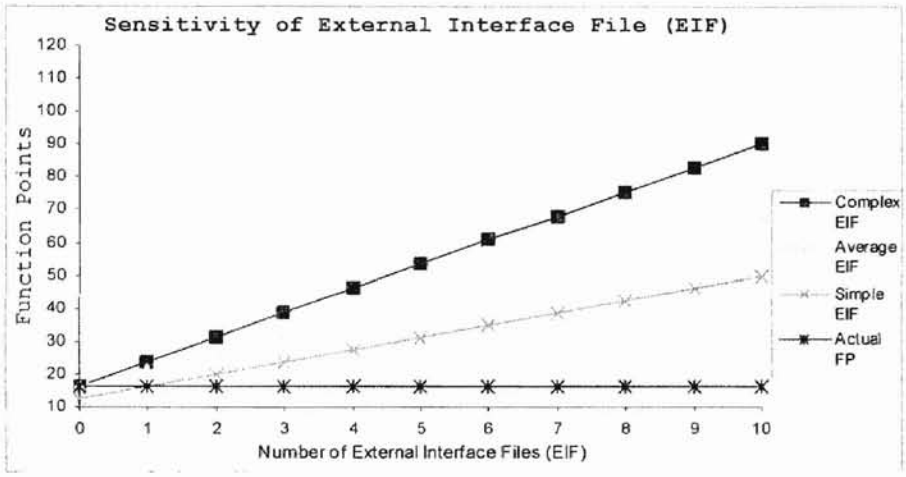

Sensitivity of General System Characteristics
General System Characteristics
0.75
0.73
0.71
0.69
0.67
0.65
0 1 2 3 4 5
Complexity Factor
1 2 3 4 5
6 7 8 9 10
11 12 13 14 Actual GSC

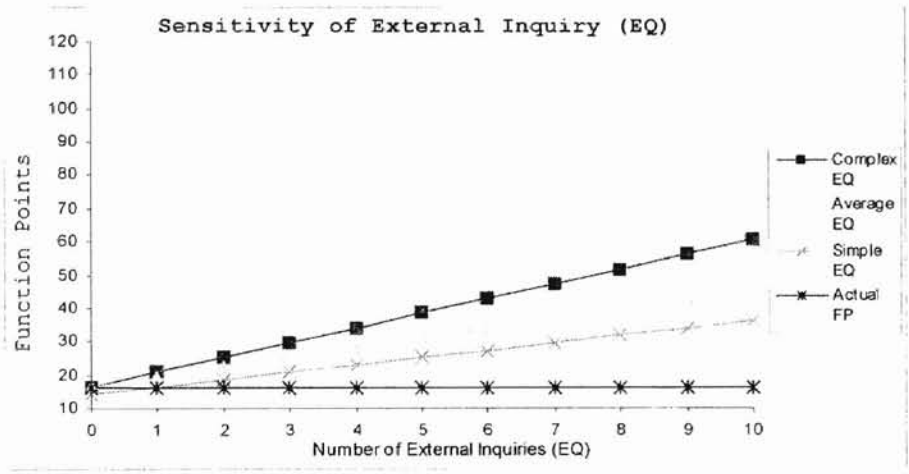
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XCALENDER PROGRAM

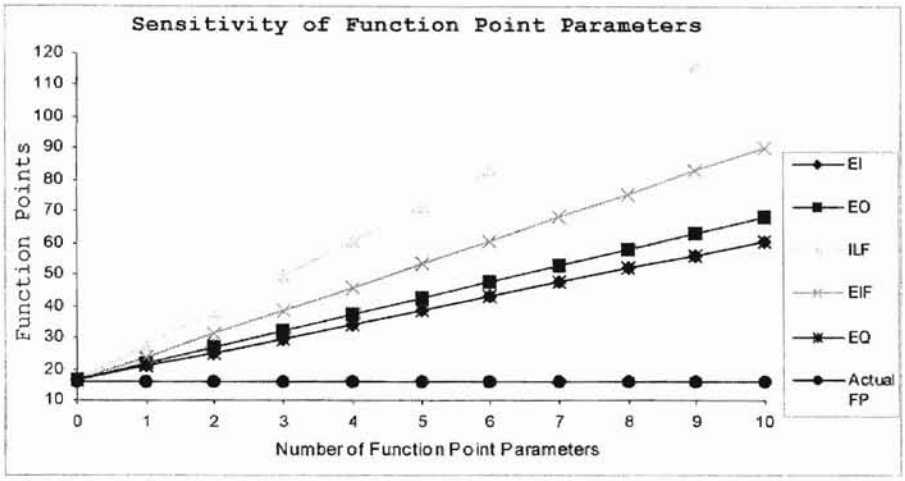


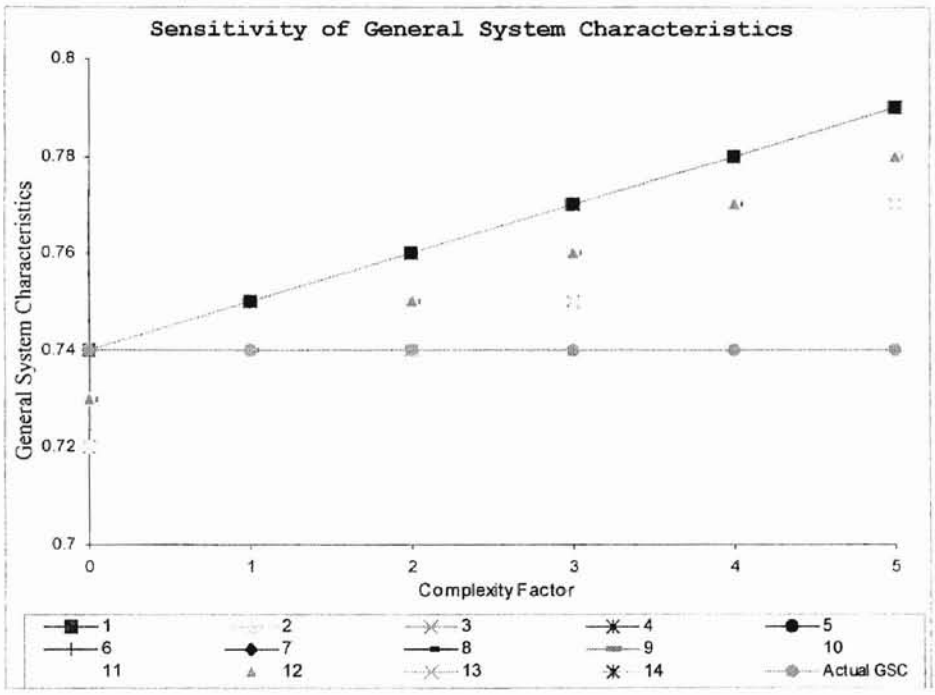



Sensitivity of Internal Logical File (ILF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

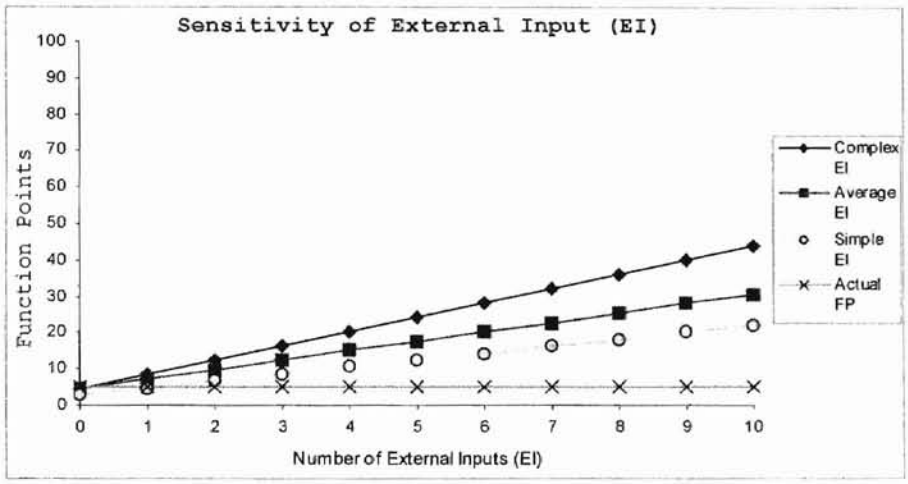

Sensitivity of External Interface File (EIF)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

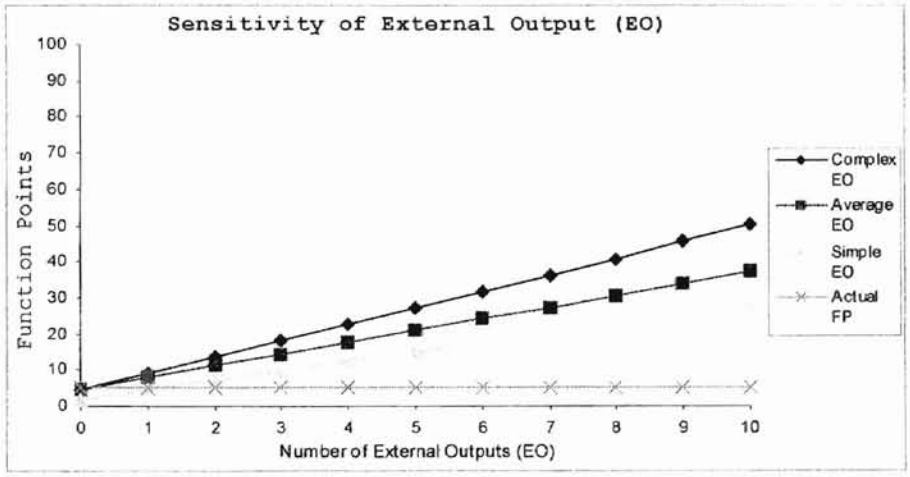

Sensitivity of External Inquiry (EQ)
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

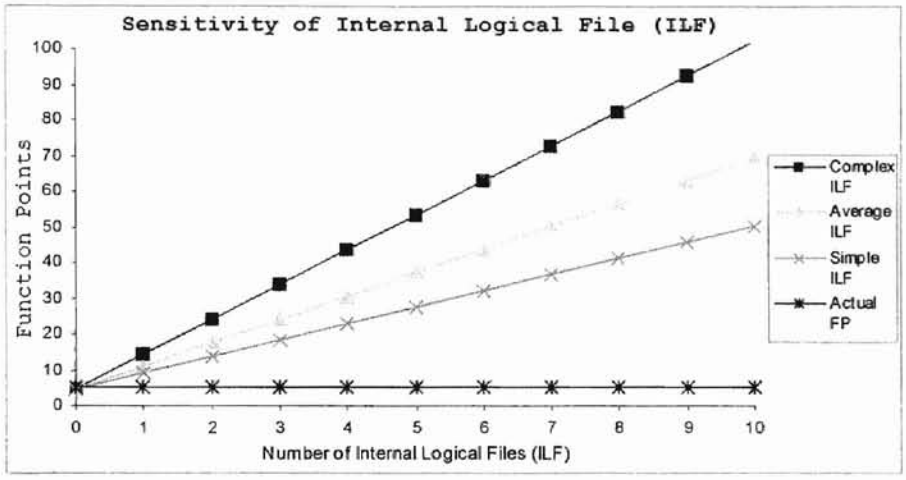

Sensitivity of Function Point Parameters
Function Points
120
110
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

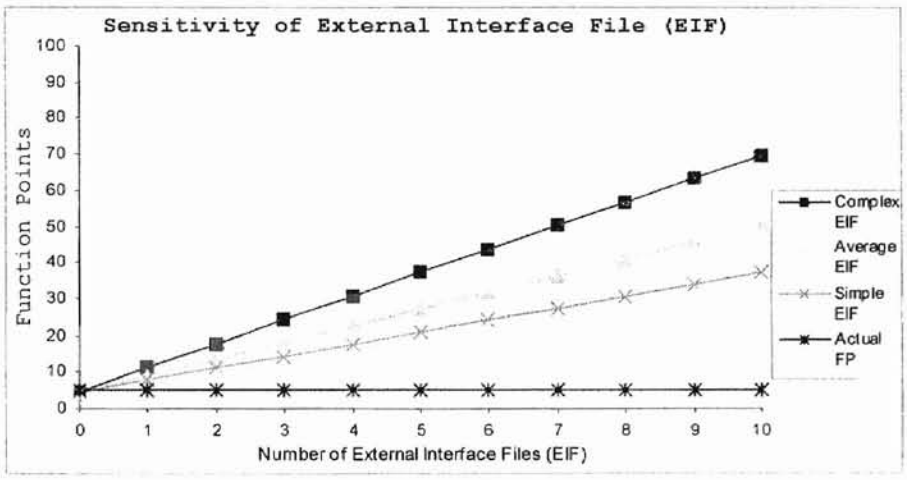

Sensitivity of General System Characteristics
General System Characteristics
0.8
0.78
0.76
0.74
0.72
0.7
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

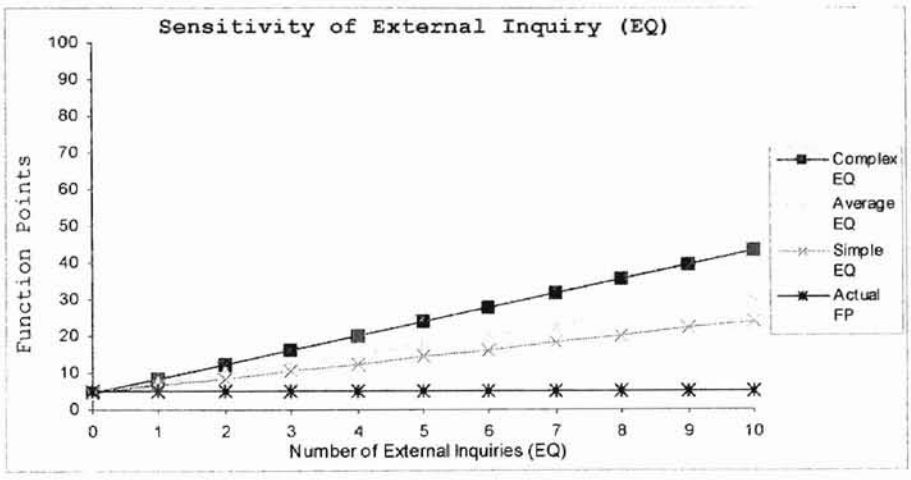
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XDL PROGRAM

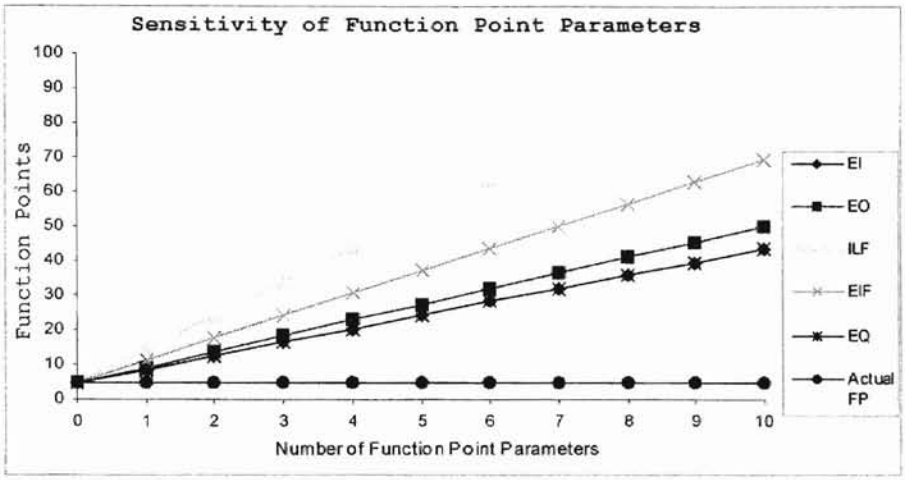


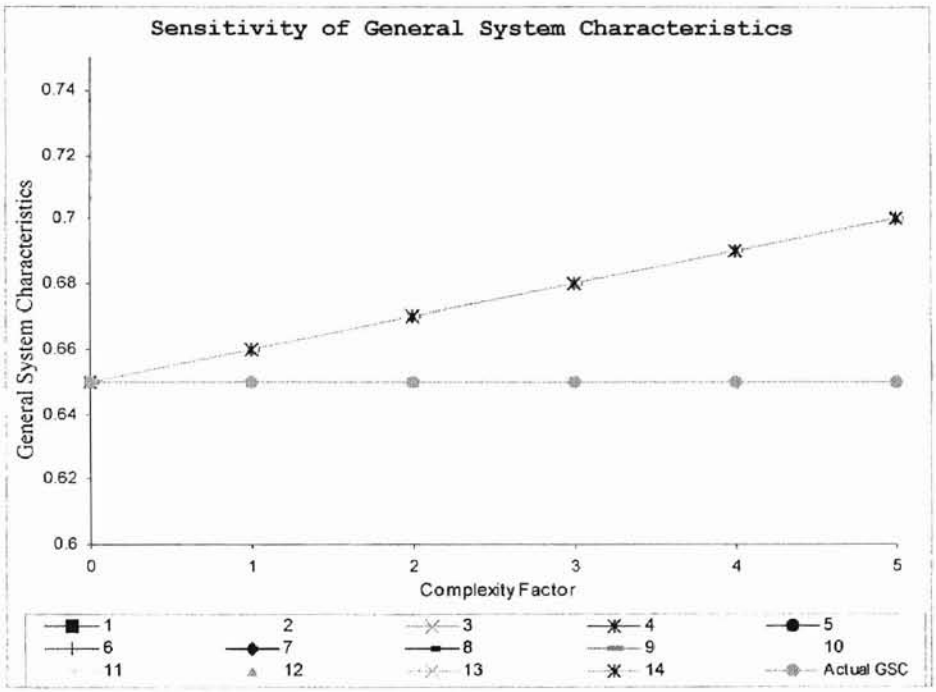



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

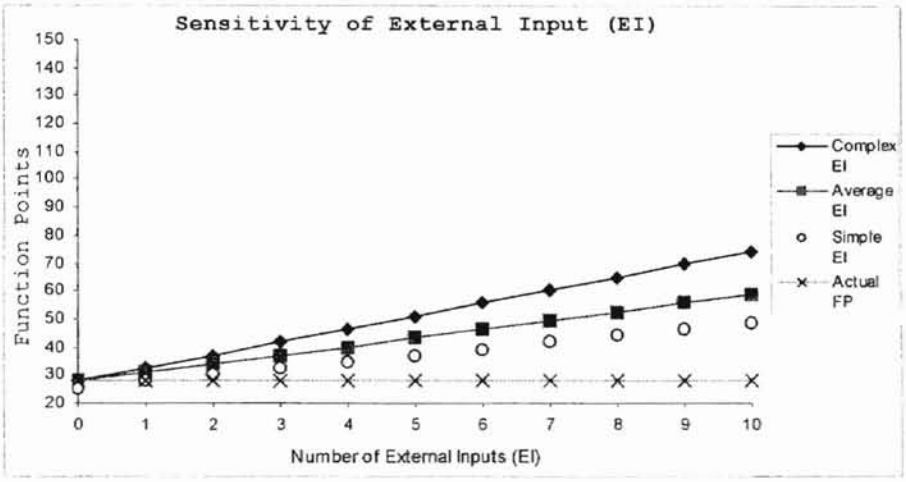

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

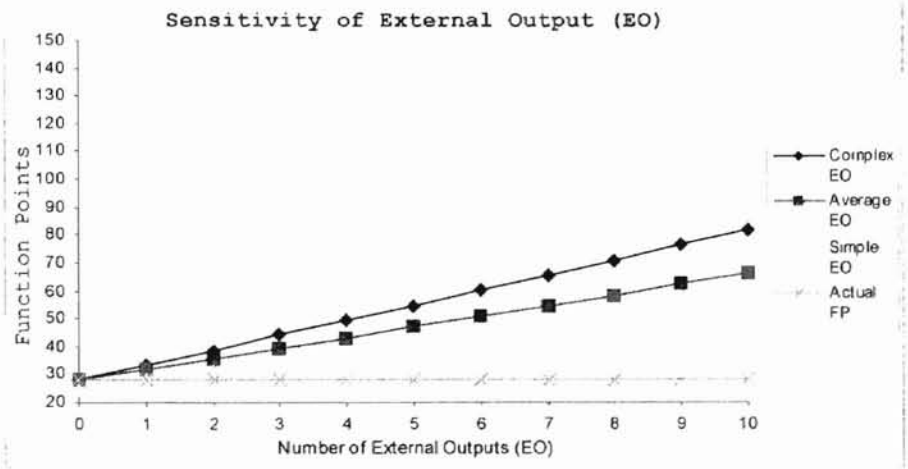

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

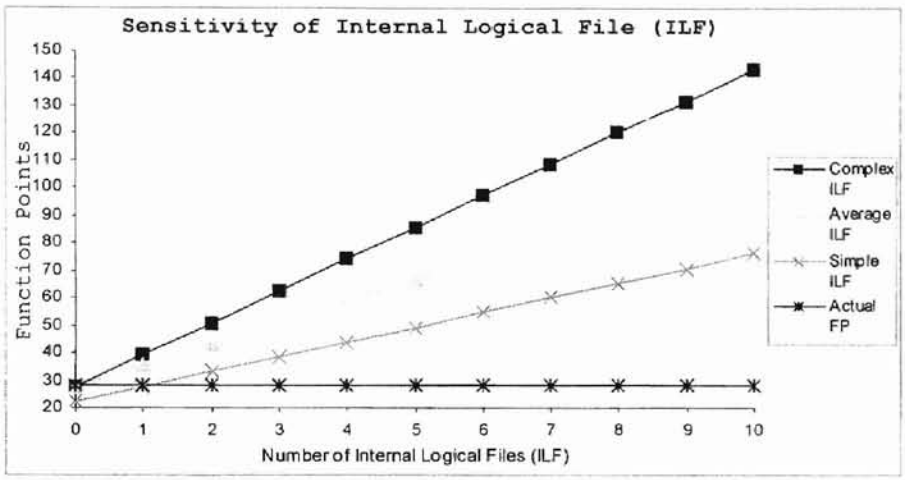

Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

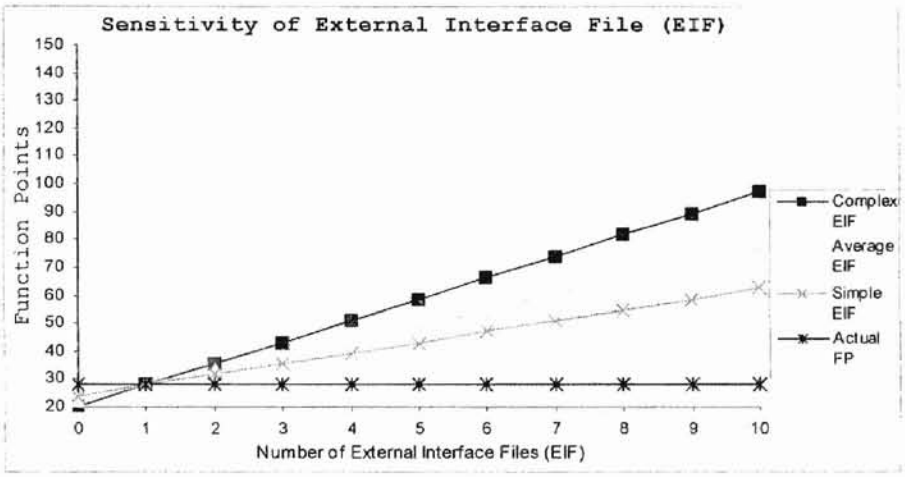

Sensitivity of General System Characteristics
General System Characteristics
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

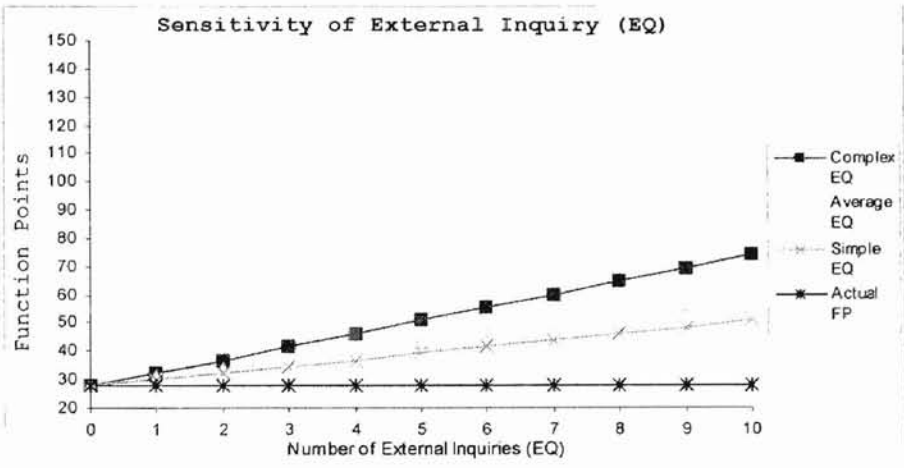
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XGETFTP PROGRAM

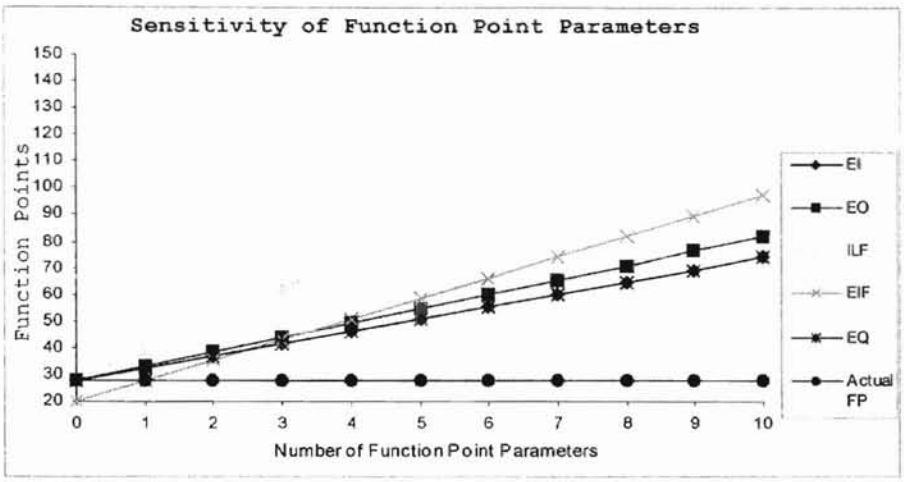


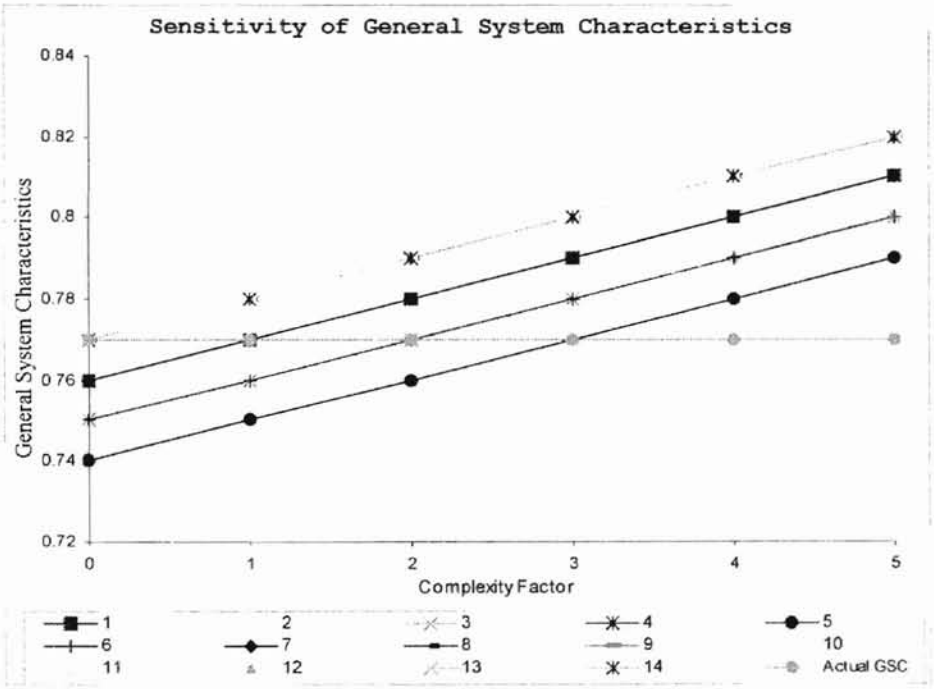



Sensitivity of Internal Logical File (ILF)
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
20
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

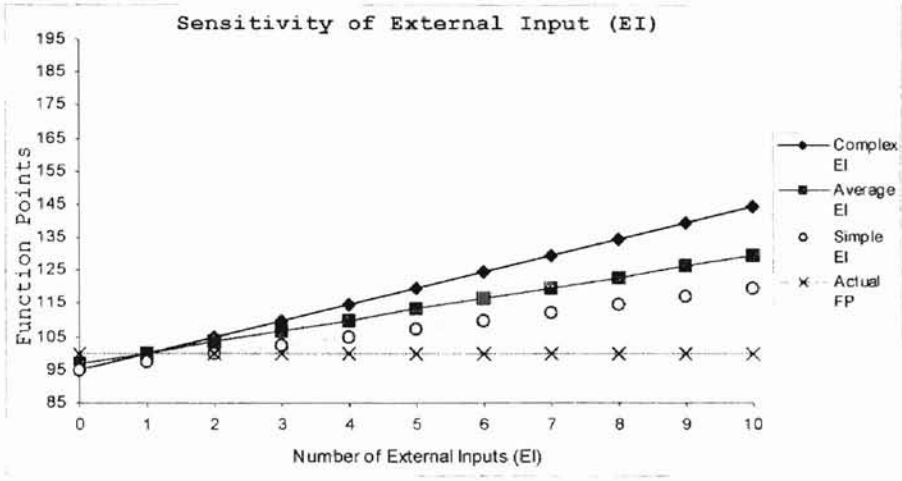

Sensitivity of External Interface File (EIF)
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
20
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

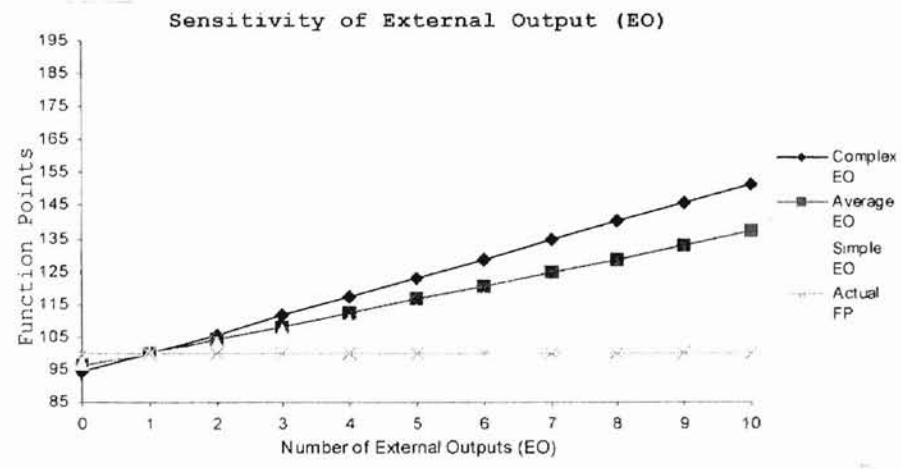

Sensitivity of External Inquiry (EQ)
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
20
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

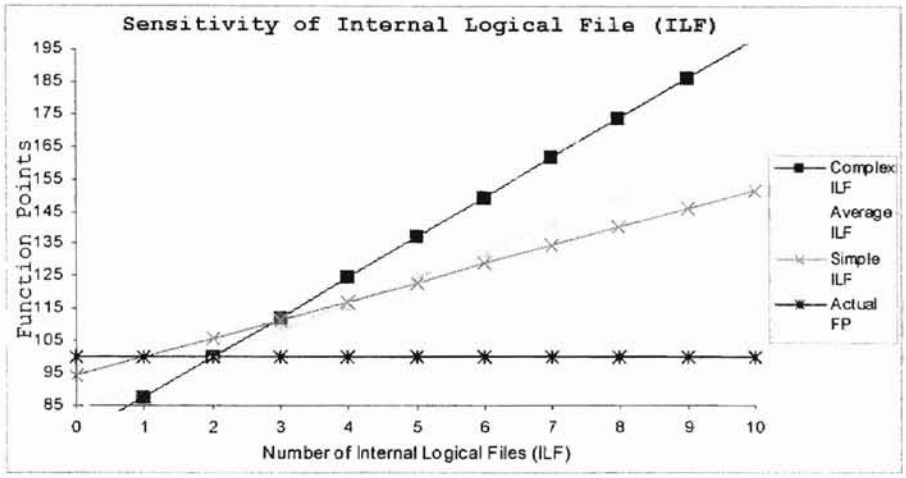

Sensitivity of Function Point Parameters
Function Points
150
140
130
120
110
100
90
80
70
60
50
40
30
20
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

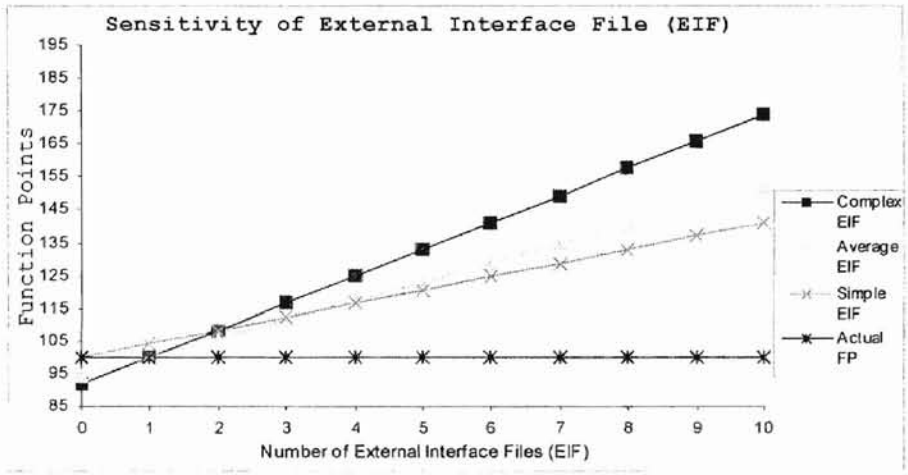

Sensitivity of General System Characteristics
General System Characteristics
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

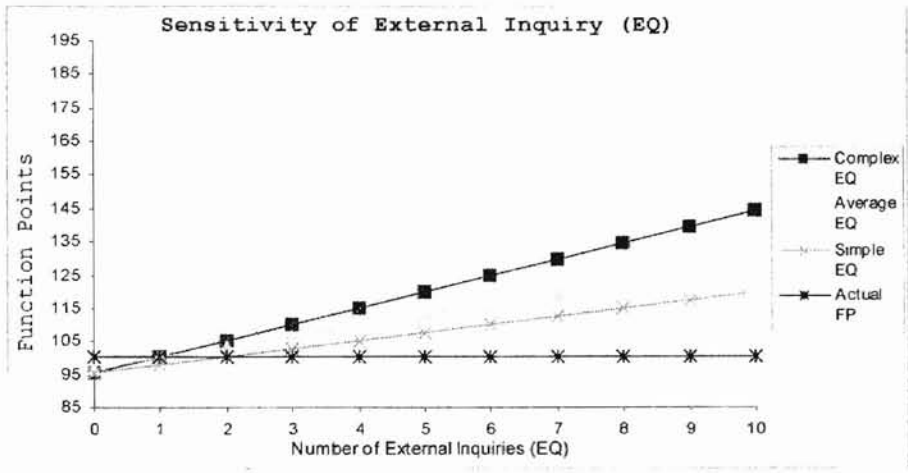
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XGOPHER PROGRAM

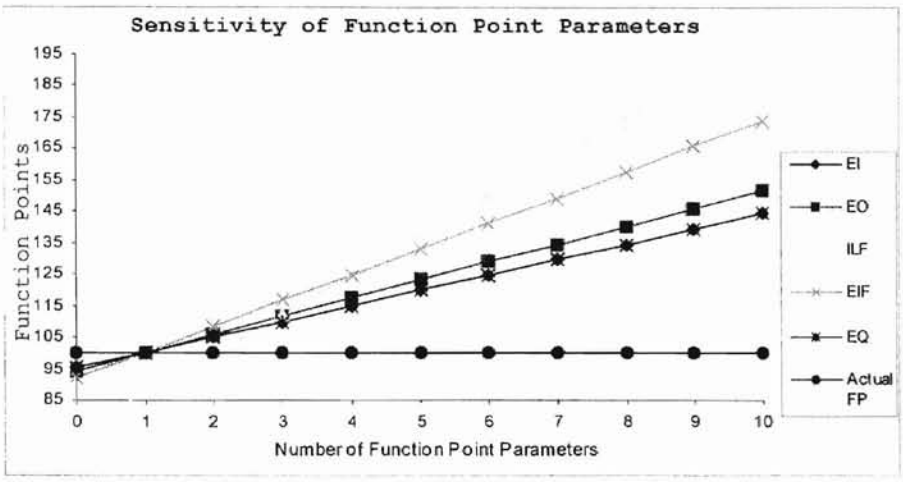


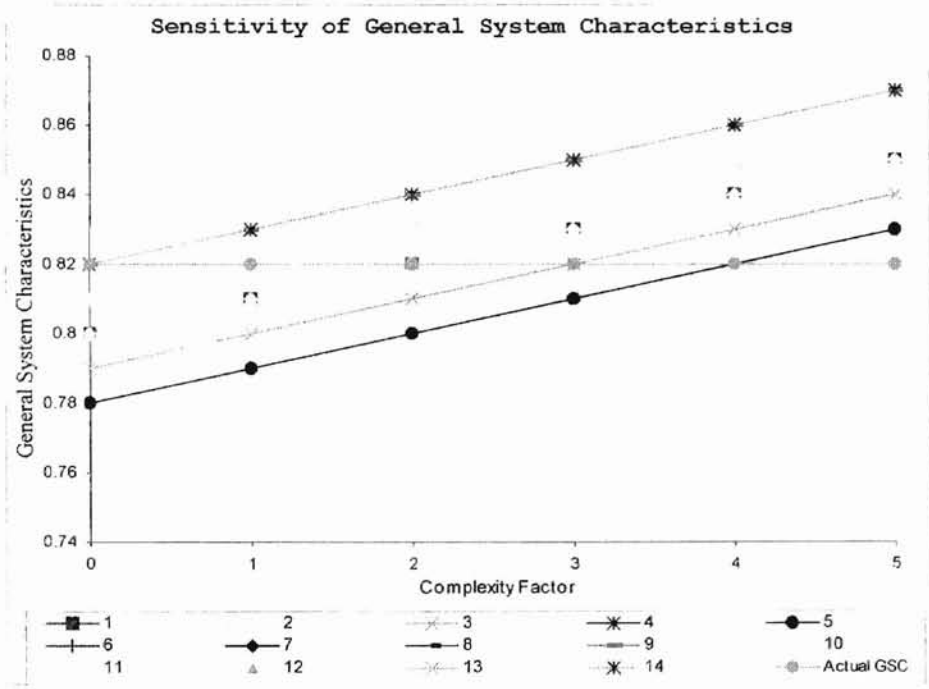



Sensitivity of Internal Logical File (ILF)
Function Points
195
185
175
165
155
145
135
125
115
105
95
85
0 1 2 3 4 5 6 7 8 9 10
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

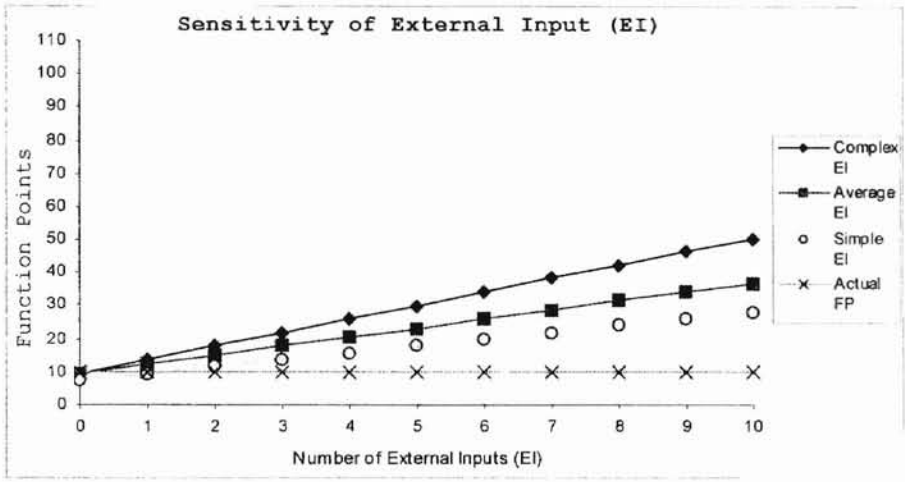

Sensitivity of External Interface File (EIF)
Function Points
195
185
175
165
155
145
135
125
115
105
95
85
0 1 2 3 4 5 6 7 8 9 10
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

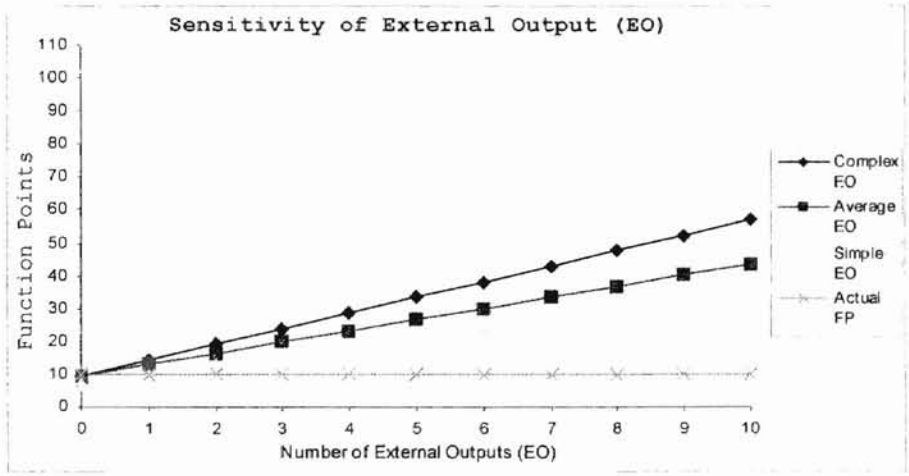

Sensitivity of External Inquiry (EQ)
Function Points
195
185
175
165
155
145
135
125
115
105
95
85
0 1 2 3 4 5 6 7 8 9 10
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

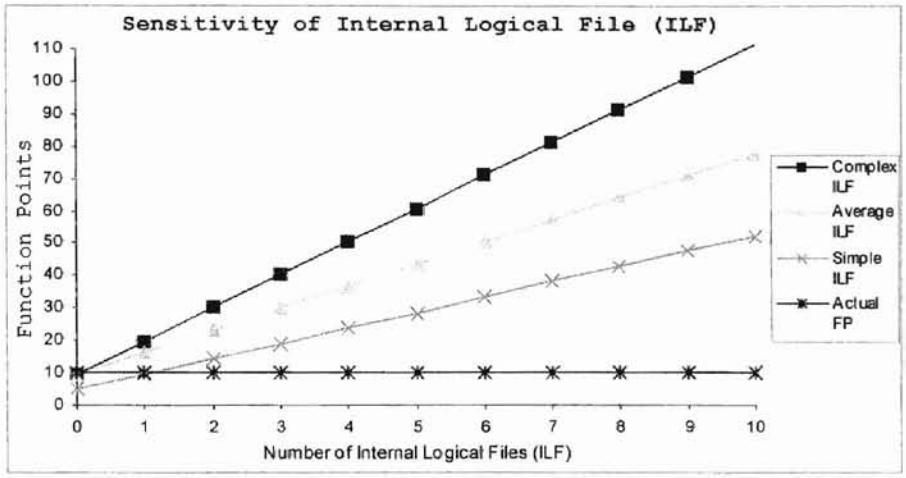

Sensitivity of Function Point Parameters
Function Points
195
185
175
165
155
145
135
125
115
105
95
85
0
1
2
3
4
5
6
7
8
9
10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

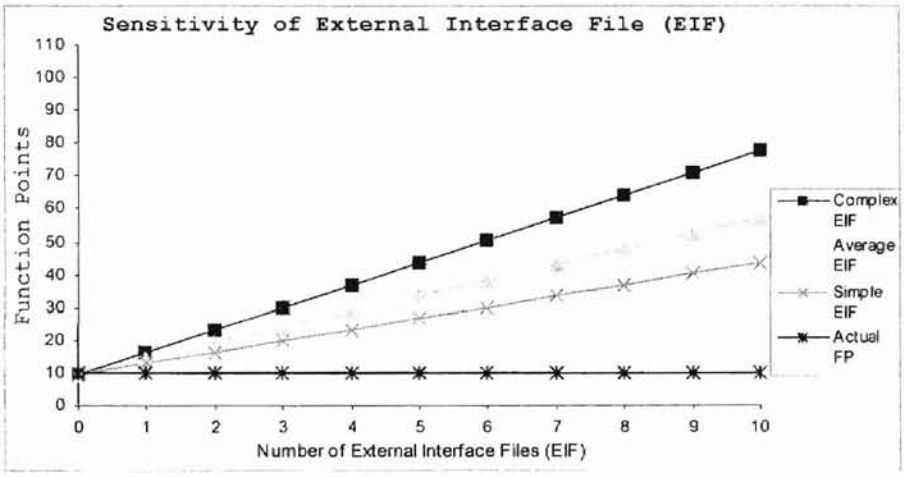

Sensitivity of General System Characteristics
General System Characteristics
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0
1
2
3
4
5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

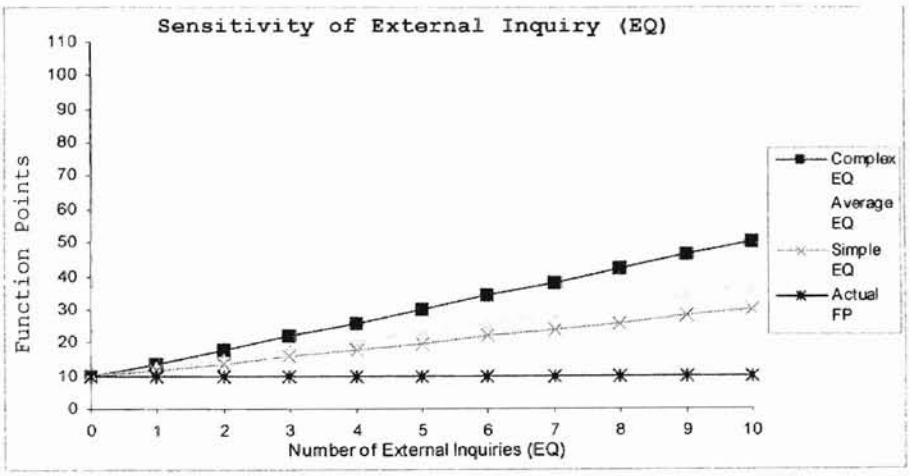
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XNLOCK PROGRAM

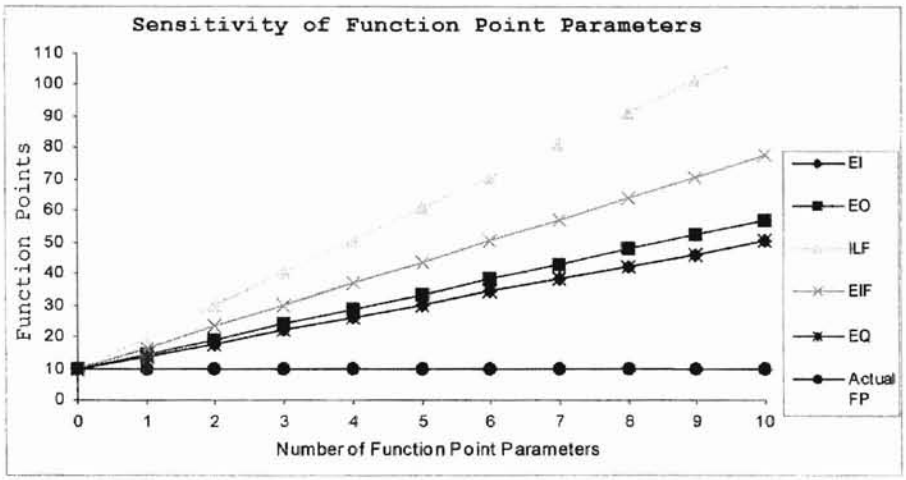


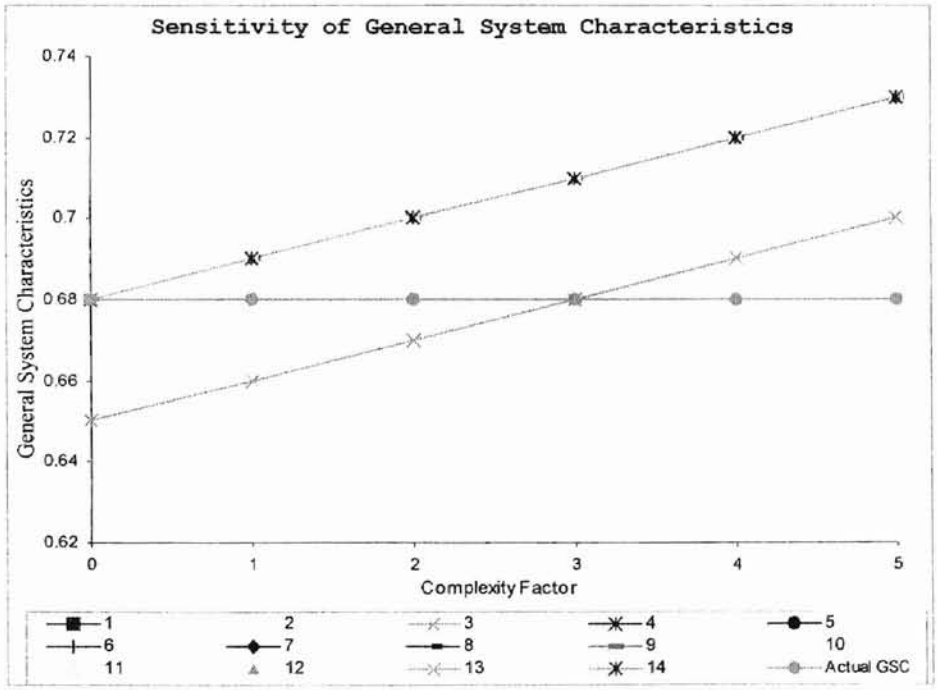



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

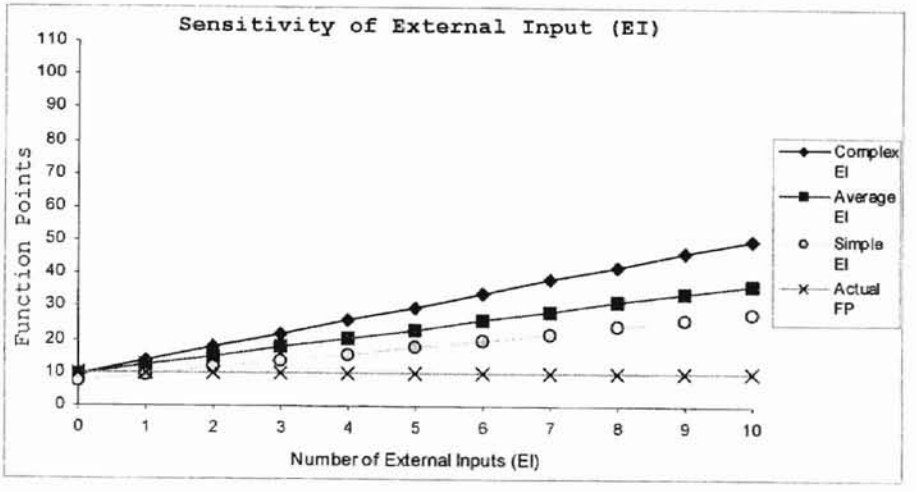

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

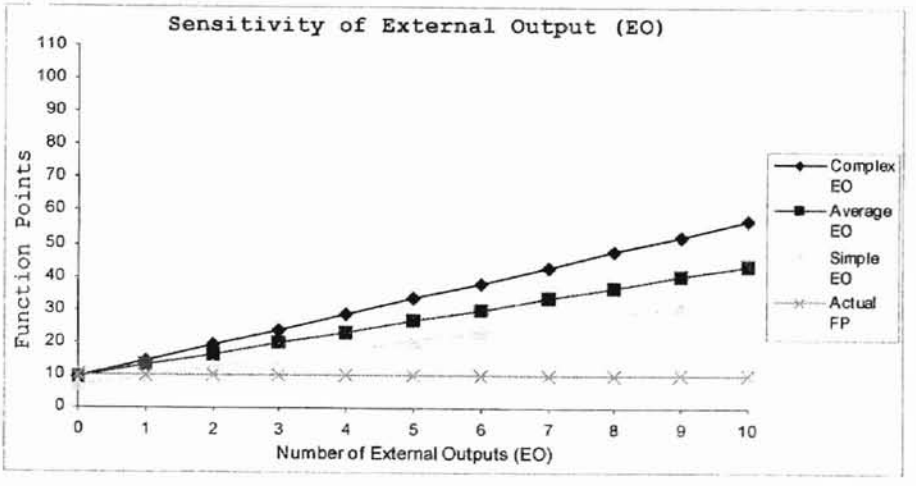

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

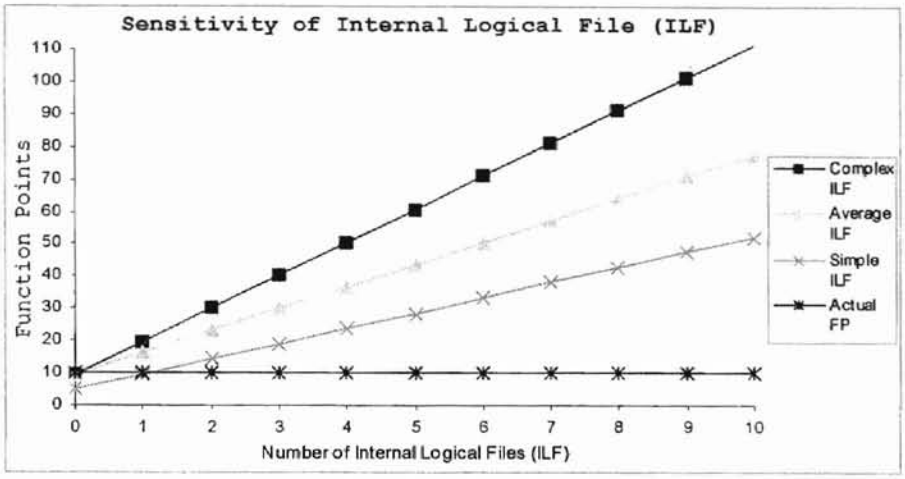

Sensitivity of Function Point Parameters
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

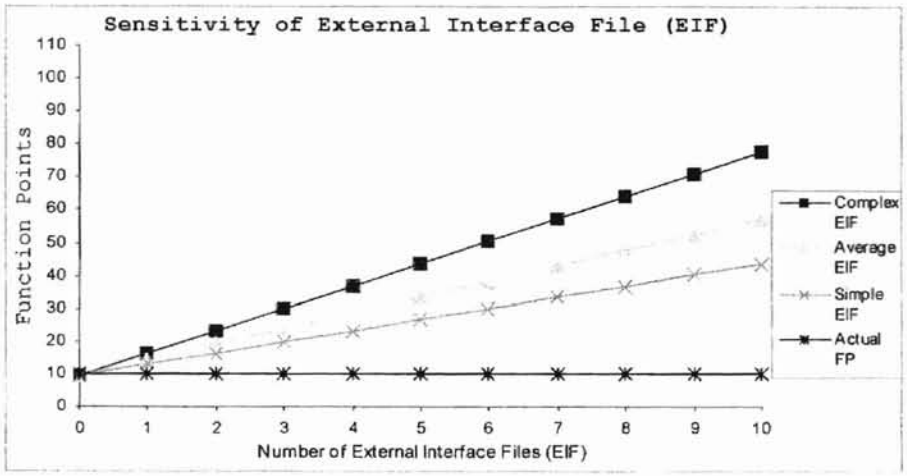

Sensitivity of General System Characteristics
General System Characteristics
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0 1 2 3 4 5
Complexity Factor
1 2 3 4 5
6 7 8 9 10
11 12 13 14 Actual GSC

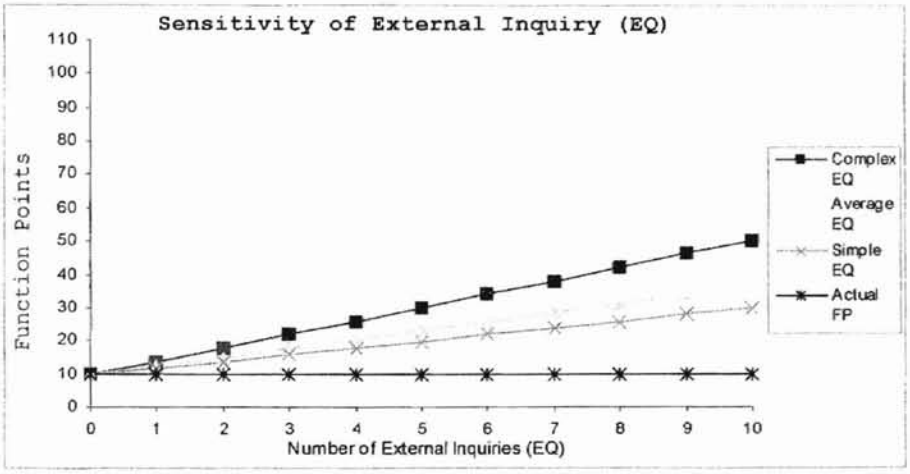
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XPOSTIT PROGRAM

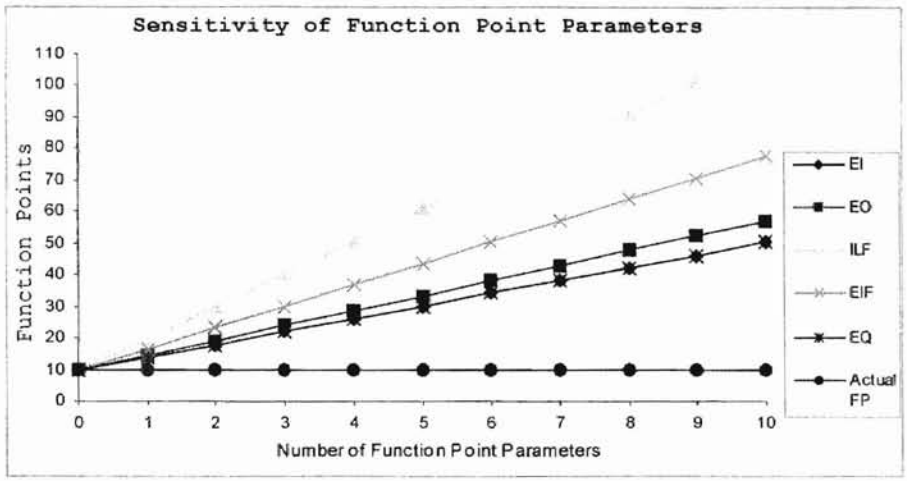


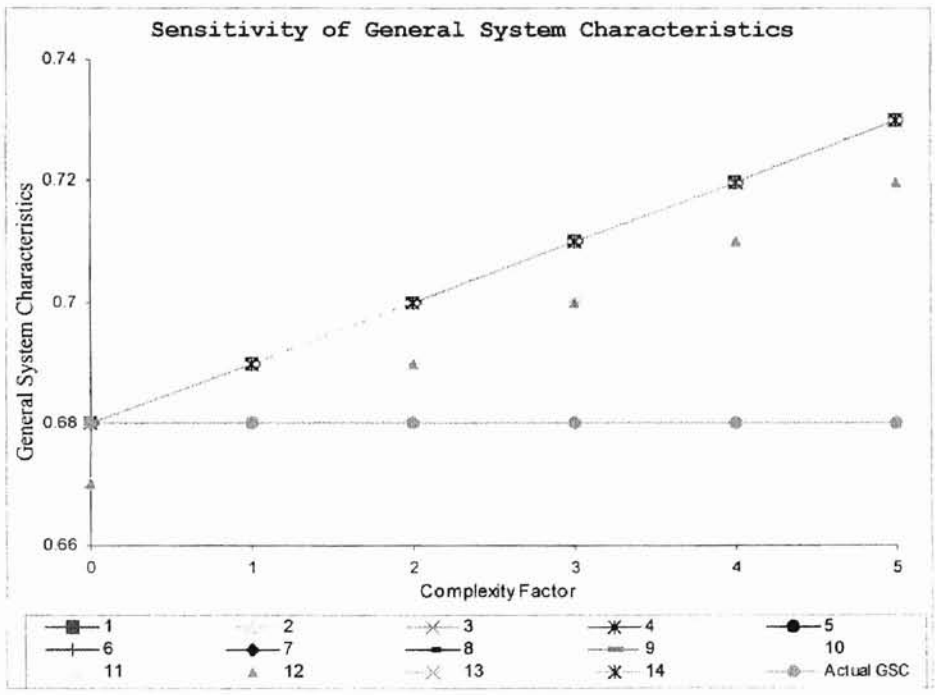



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP

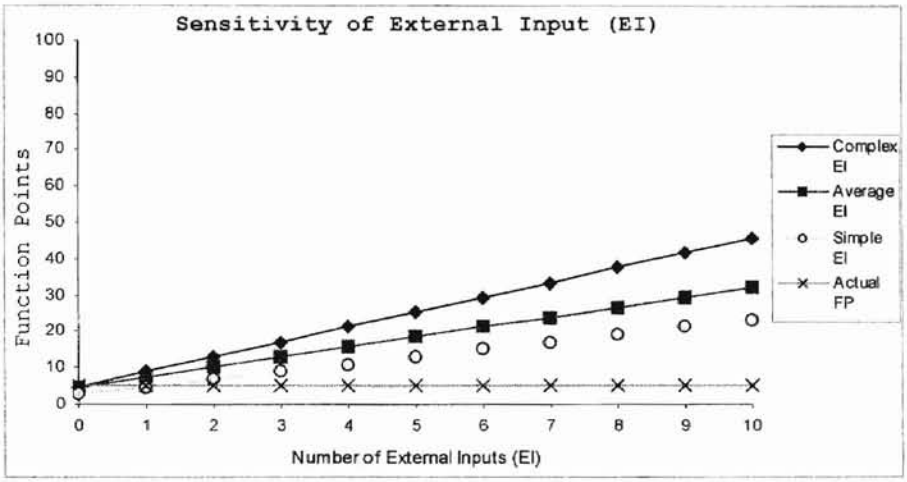

Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP

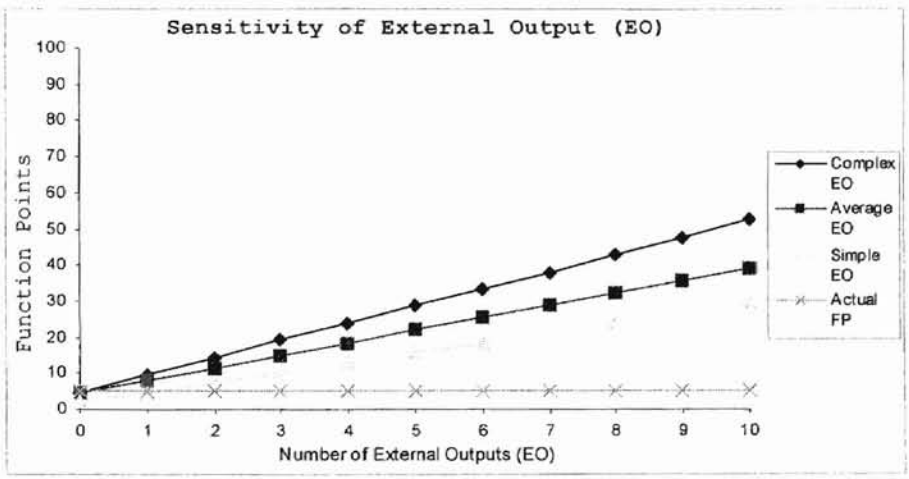

Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP

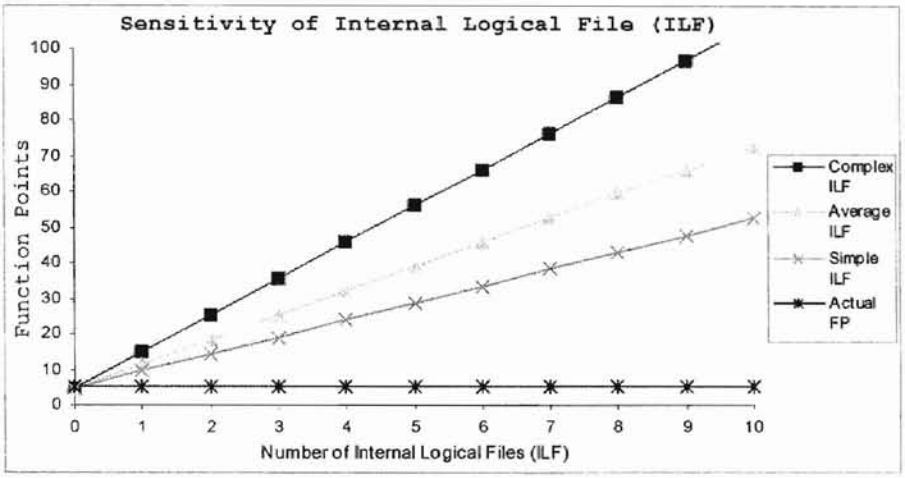

Sensitivity of Function Point Parameters
Function Points
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP

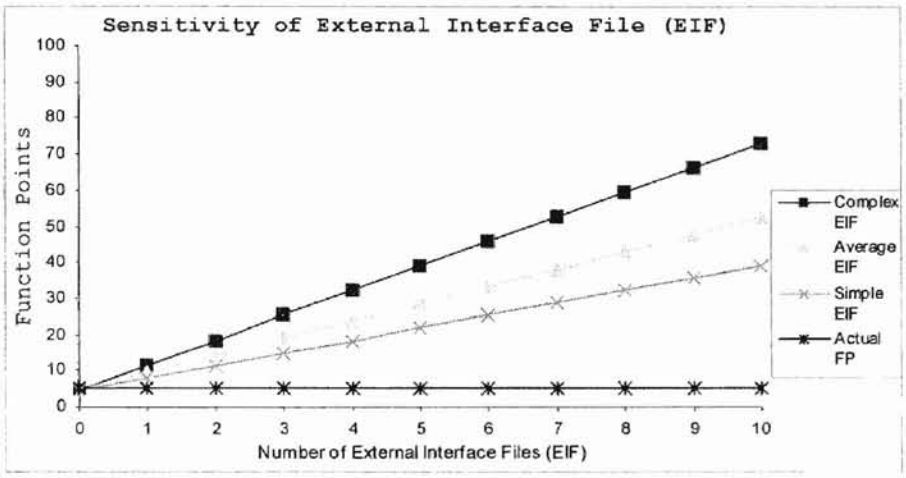

Sensitivity of General System Characteristics
General System Characteristics
0.74
0.72
0.7
0.68
0.66
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

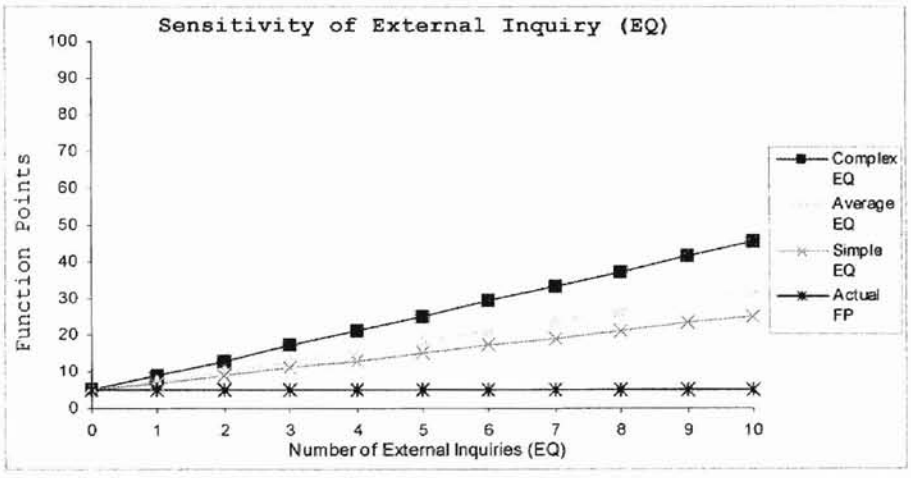
SENSITIVITY ANALYSIS OF FUNCTION POINT METRIC FOR THE XTALK PROGRAM

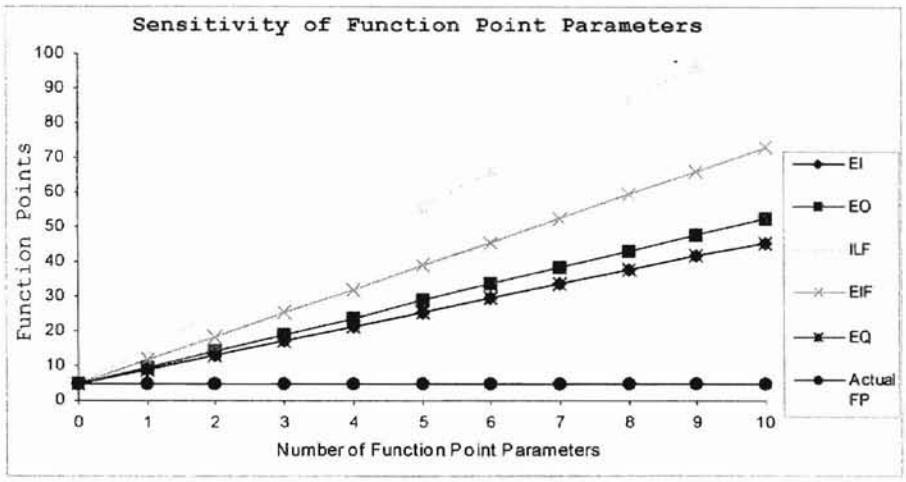


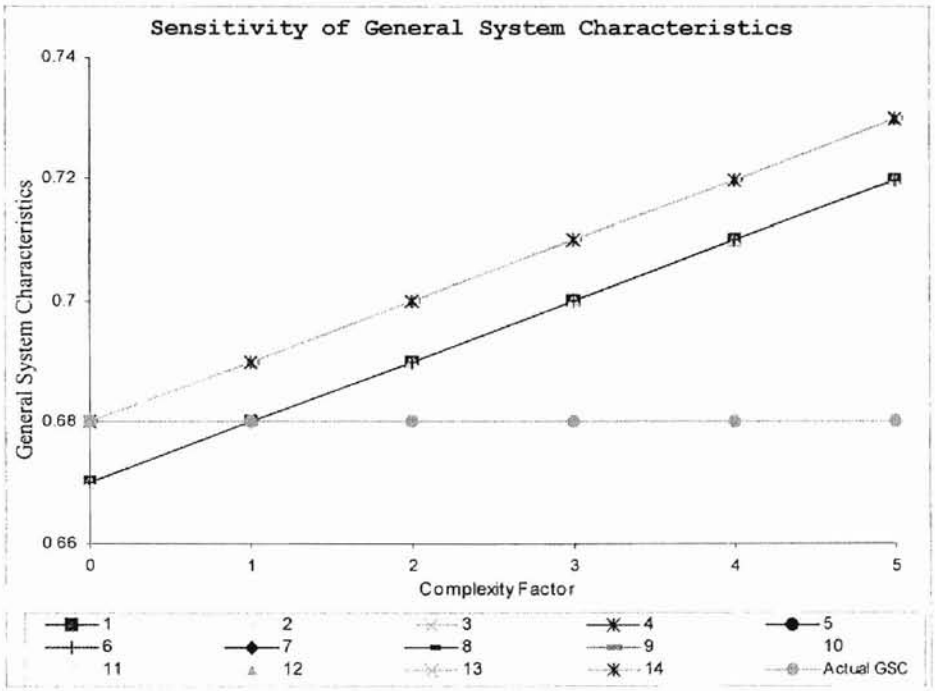



Sensitivity of Internal Logical File (ILF)
Function Points
Number of Internal Logical Files (ILF)
Complex ILF
Average ILF
Simple ILF
Actual FP


Sensitivity of External Interface File (EIF)
Function Points
Number of External Interface Files (EIF)
Complex EIF
Average EIF
Simple EIF
Actual FP


Sensitivity of External Inquiry (EQ)
Function Points
Number of External Inquiries (EQ)
Complex EQ
Average EQ
Simple EQ
Actual FP


Sensitivity of Function Point Parameters
Function Points
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10
Number of Function Point Parameters
EI
EO
ILF
EIF
EQ
Actual FP


Sensitivity of General System Characteristics
General System Characteristics
0.74
0.72
0.7
0.68
0.66
0 1 2 3 4 5
Complexity Factor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Actual GSC

VITA

Tongchit Tantikul

Candidate for the Degree of

Master of Science

Thesis: FUNCTION POINT METRIC CALCULATION AND SENSITIVITY ANALYSIS

Major Field: Computer Science

Biographical:

Personal Data: Born in Bangkok, Thailand, May 28, 1971, daughter of Mr. Chitporn Tantikul and Mrs. Valaivan Tantikul.

Education: Received Bachelor of Science degree in Statistics from Chulalongkorn University, Bangkok, Thailand, in April 1993; completed the requirements for Master of Science degree in Computer Science at the Computer Science Department at Oklahoma State University in May 1998.

Professional Experience: Teaching Assistant, Computer Science Department, Oklahoma State University, August 1997 to May 1998; Computer Lab Consultant, Computing and Information Services, Oklahoma State University, October 1996 to August 1997.